FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Hashtag popularity prediction for social networks

Ivo Lima da Silva

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Luís Filipe Pinto de Almeida Teixeira

March 1, 2018

Hashtag popularity prediction for social networks

Ivo Lima da Silva

Mestrado Integrado em Engenharia Informática e Computação

March 1, 2018

Abstract

Currently, in social networks, labeling media content with hashtags is used to make the content reachable to more visualizations. The ones who do use this marketing method to promote their products, vary from big companies like worldwide brands or individuals, such as freelancers or photographers, since it usually involves low costs.

However, a real time analysis of the social networks' hashtags' popularity ecosystem is a complex task, whose efficiency is hard to evaluate. This is due to the nature of hashtags: they can have abstract meanings, created at any time by any user. The implemented system addresses this problem since it follows the behaviour of a set of hashtags and measures the impact each hashtag as on the posts that use it.

Instagram is the platform of choice for users who share content with hashtags because Instagram does not punish its users for the use of many hashtags, like many other Social Networks do. This and hashtag specific features, like search and follow hashtags, made Instagram the Social Network we chosen for our implementation.

To analyse the behaviour of a set of hashtags, we implemented a web scraper module that made possible to fetch information from Instagram. With this module, we perform a graph search to find different hashtags and understand how they were related.

This graph search resulted in 17599 hashtags (vertices) and 41777 connections (edges) between hashtags. With that result, we used Walktrap [PL06] and Pagerank [PBMW99] to find what hashtags were more relevant and less abstract.

For that set of hashtags the system autonomously analysed the posts that used them, checks their likes' progress through time. When a post reaches its end of life, it adds a new popularity value (popularity index) to the hashtags it uses related with the expected and real likes of the post.

Having a sequence of each hashtag's popularity indexes, we then implemented a Neural Network with LSTM architecture. We fine tuned the hyperparameters of the NN which resulted in a Mean of each hashtag prediction's MSE of *0,05488*.

The results show that it is able to predict the near future behaviour of a hashtag popularity index. Understading the peak relevance of a hashtag allows users to take advantage of the most relevant hashtags. This allows users to schedule their publications or choose the most popular hashtags for their media content.

This tool opens the possibility for users to have a bigger influence in Social Networks.

Resumo

Atualmente, nas redes sociais, são usadas *hashtags* para tornar conteúdo midia mais acessível a mais visualizações. Quem usa este método the marketing para promover os seus produtos, varia desde empresas grandes, tais como marcas internacionais, ou indivíduos, tais como *freelancers* ou fotógrafos, dado que este método não envolve custos avultados.

No entanto, uma análise em tempo real do ecosistema das *hashtags* popularidade numa rede social, é uma tarefa complexa. Isto é devido à natureza das *hashtags*: podem ter vários significados e podem ser geradas a qualquer altura e por qualquer utilizador. O sistema implementado aborda este problema, acompanhando o comportamento de um conjunto de *hashtags* e mede o impacto que cada *hashtag* tem nas publicações que a usam.

A rede Instagram é a plataforma de escolha para utilizadores que partilham mídia com *hashtags*. A razão deve-se ao facto do Instagram não penalizar os utilizadores que usam muitas *hashtags*, tal como outras redes sociais fazem. Adicionando várias funcionalidades que ajudam o uso de *hashtags*, tais como pesquisa e "seguir" *hashtags*, fizeram com que o Instagram fosse a rede social escolhida para a nossa implementação.

Para analisar o comportamento de um conjunto de *hashtags*, foi implementado um *web scraper* para ser possível arrecadar informação do Instagram. Com este módulo, foi feita uma pesquisa em grafo para encontrar *hashtags* diferentes e perceber como estavam relacionadas.

A pesquisa em grafo resultou em 17599 *hashtags* (vértices) e 41777 relações (arestas) entre *hashtags*. Com esse resultados, foi usado o algoritmo Walktrap [PL06] e Pagerank [PBMW99] para descobrir quais *hashtags* eram mais relevantes e menos abstractas, que depois iriam ser acompanhadas.

Para o conjunto de *hashtags* a ser acompanhado, o sistem autonomamente analisava as publicações que possuiam essas hashtags e verifica o progresso dos seus *likes* ao longo do tempo. Quando uma publicação chega ao fim da sua vida relevante, o sistema adiciona um novo índice de popularidade às *hashtags* que a publicação usa. Esse índice de popularidade está relacionado com o valor de *likes* esperado e o valor real de *likes*.

Tendo uma sequência de índices de popularidade para cada *hashtag*, foi implementado uma Rede Neuronal de arquitectura *Long Short-Term Memory*. Depois de afinados os hiper-parâmetros da Rede Neuronal, e tendo calculdado o MSE para cada *hashtag*, a média dos MSEs foi de *0,05488*.

Os resultados mostram que o model é capaz de prever o futuro próximo do índice de popularidade de uma *hashtag*. Compreender o comportamento da popularidade de uma *hashtag*, permite aos utilizadores tirar proveito das *hashtags* mais relevantes. Isto permite aos utilizadores planear quando fazer as suas publicações ou escolher quais as *hashtags* mais populares para a sua mídia.

Este sistema abre a possibilidade para os utilizadores terem uma influência maior nas Redes Sociais.

"Good is the enemy of great.
And that is one of the key reasons why we have so little that becomes great.
We don't have great schools, principally because we have good schools.
We don't have great government, principally because we have good government.
Few people attain great lives, in large part because it is just so easy to settle for a good life."

James C. Collins

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

API	Application Programming Interface
ARIMA	Autoregressive integrated moving average
IAPI	Instagram's API
LSTM	Long Short-term Memory
MSE	Mean Squared Error
NN	Neural Network
OSS	Open Source Software
RNN	Recurrent Neural Networks
SEO	Search Engine Optimization

Chapter 1

Introduction

In this chapter, it's provided an overview of this thesis belongs. First, it's presented an explanation of the context of this work then the identification of the problem intended to solve and the underlying motivations. It's also described the vision for the complete system and delineated the focus of this thesis.

1.1 Context

Social networks, more specifically online ones are platforms that are used by people to connect to other users and to share media content such as photos, videos or text-based messages. In spite of its large growth in early 2000s, the first major online social network was launched in 1997 [How08].

> "User-generated content is the lifeblood of social media.
> Web 2.0 is the ideology and user-generated content is the fuel." [OW15]

User-generated content is any type of digital content in the form of photos, videos, blogs, discussion topics or audio that is created and uploaded by users and that can be publicly available to other users. This type of content is king in social media, so online social networks cannot subsist without it. It is this type of content that can define the quality of the platform. For example, many social media websites cannot survive due to the lack of user-generated content. Having a large user base is not enough, for example MySpace was one of the most used Social Networks which lost its relevance due to the lack of content and new competitors.

Due to the large user base of social networks, companies see an opportunity to reach a bigger audience and look at social networks' potential for advertising. This was also the business model of the social networks since its use is free. According with [Ste17], an overwhelming majority of marketers (93%) use Facebook Ads, followed by Instagram Ads (24%) in second place. Even with such big numbers, many marketers intend to increase their activity on paid ads in Social Networks.

However, with the popularity of ad blocking tools, many companies are changing their strategy. Instead of direct ads, companies use *influencers* to reach a larger audience due to their loyal followers.

Influencers are popular people that mantain high activity on social networks that often get sponsored by companies to promote their products in social networks. *Influencers* often take advantage of trends to boost their popularity/virality.

> "Virality: individuals create it, governments fear it, companies would die for it." [NH13]

Due to the nature of the business, *influencers* can make profit from this kind of life, generating more followers and more revenue to the companies that sponsor them. They specialize in communication and brand promotion and are generally associated with specific life styles. *Influencers* are a great way for companies to get their products known by the public without being too intrusive.

Despite the big base of followers, *influencers* can always do an effort to grow their audience. One way of doing this is by using hashtags to label their social content. Hashtag is a word or phrase prefixed by the symbol *#*. It's a technological labeling mechanism supported by the majority of social networks. Originally, it was created as "channel tags" for users to join a particular discussion [Mes07] but the hashtag has been notably key in publicizing social issues [BB11]. With the strategic use of hashtag, media content can be broadcast to broader audiences. Hashtags also help organize and categorize information for *search bots*, which improves SEO (Search Engine Optimization). Hashtags in social networks can be divided into two groups:

Community hashtags are the ones that connect like-minded users around a specific subject or movement. Community hashtags improve SEO of social media and grow communities.

Figure 1.1: Hashtag *#meToo*

Branded hashtags These hashtags represent something unique to a business or brand. It can be simply the company name, catchphrase or the company's product name.

Figure 1.2: Hashtag *#ShareACoke* by Coca-Cola

While community hashtags are meant to increase the reach of a message and find people with similar interests, branded hashtags are used to connect themes for brands and their audience. They can also be used to promote a campaign or aggregate user-generated content.

As [Mav17] states,

> "User-generated content featuring a brand drove 6.9x higher engagement than brand-generated content."

which means that brands that prefer using community hashtags achieve better recognition and engagement.

1.2 Motivation

Using hashtags as a tool for social media marketing is not as simple as it seems. To know how to use suitable hashtags and in the right context is an analysis that requires an evaluation. For example, a hashtag can be very popular at a given time, but if used in a publication later, the peak relevance of the post can be reached when the hashtag turn to be irrelevant. Since many social networks impose a hashtag limit to each publication, it's difficult to guarantee that the chosen hashtags will be relevant to the success of the publication. So its use becomes inefficient and the users miss the opportunity to better promote media content.

To discover popular hashtags and predict their relevance in the near future, big companies can dispose the time and resources to discovering these popular hashtags. They can work their advertising plans around it and create social media especially to a trend of that time.

However, small businesses do not have the resources to use this kind of marketing strategy, since many times they are limited to what their product represent. Traditional content creators like photographers and videographers that try to sell their media by taking advantage of trends and reaching out to their niche market, are good examples of those cases. This means that small businesses don't have funds to find out trends and fit to their product since their only strategy is to work around their product. For that, there is a need for an automated tool that analyses their content and returns the proper hashtags to that context and provides a metric for the popularity effect that each hashtag will have.

Consider that, we are not looking toward to if an hashtag is popular or not, - which can be translated by counting how many times that hashtag is searched or even with other metrics - but the effect that the hashtag has on social media where it's used.

The aknowledgement of the hashtag ecosystem's current state is important in itself, but there's even more relevance in understanding the evolution of the popularity effect of a hashtag. If the evolution is known, then it's possible to predict the near future of the popularity effect. This is particularly important because on social media no action is instantaneous. Every hashtag on social media is relevant for some periods of time, depending on trends. So it can be inefficient if a hashtag is used when it is not in its popularity peak. By using hashtags in these conditions, the media post does not reach the popularity potential that its hashtags one used to achieve.

1.3 Proposed system

In order to solve this issue, it was idealized a system that would recommend hashtags based on the media content. This system would also give a numeric context to each hashtag in order to provide information about the future of the popularity effect of a given hashtag.

The system is branched in two components:

Content analysis that would analyse the content of a given media object and classify the media with a set of hashtags;

Popularity prediction that would follow the evolution of the media that uses a pre-defined set of hashtags and be able to predict the future values of popularity effect of a hahstag.

1.3.1 Content analysis

This component would analyse the content of a soon to be media object. This object could be in any type of content, for instance text, photos or video.

In text based media, this could be accomplished by using natural language processing, more specifically semantic analysis.

From the data processing point of view, semantics analysis provide context to language. It provides clues not only to the meaning of words, but to their correlation with other words and other expressions. The goal, just like the human brain processes, is not to look at the meaning of a single words on the text but to see the meaning of the whole content.

In photos or video content, we could make use of computer vision. Computer vision is the field that deals with how computers can reach high-level understanding of digital images. This field's goal seeks to recreate the automated tasks that the human visual system can process, regarding image classification. Computer Vision goes beyond image classification and deep into methodology for image acquisition, processing and analysing. In a system like the proposed one, the acquisition process is already done, since the images are considered a input of the system.

Content analysis would consist of the well studied problem of image classification since image processing, in this case, would be used to help image classifcation (segmentation, color and exposure processing, etc).

1.3.2 Popularity prediction

Defining the popularity of a hashtag is not a straightforward task and might have different meanings once it depends on different points of view. Some interpretations assume that the popularity of a hashtag is related to the amount of people or media content that uses that hashtag.

In this thesis, we approach a different stance on this matter. The interpretation followed suggests that a hashtag's popularity is the efect that the hashtag has on the media content, in shape of user interaction, visualizations, likes, *re-tweets*, comments, ... This approach is complex because it's difficult to measure which component has the power of increase or decrease the popularity of

a media object. It may be caused by better content, time of publication, hashtags used or even other unknown variables. It's also difficult to measure if there was in fact a higher popularity than expected because there isn't a way to know what's the expected value.

A formula for that expected value of popularity is explored in Chapter 4.

With the values of popularity for each hashtag through time, the system must be able to predict its future values. This problem fits into the explored problem Time Series Prediction which is explored in Chapter 2.

This component, in his last form, would also be autonomous in the way that it would find new hashtags and add them to the set of hashtags to be analysed. This means that the used algorithm to predict the time series, would also need to be re-trained periodically. A system like this can have several scalability issues, like storage size and computing power, however these aspects are out of the scope of this thesis.

1.4 System to be implemented

Due to the large scope of the complete proposed system, this thesis will only focus on the Popularity Prediction and the aspects that define what is in fact popularity.

Since this project relies on media content that is labeled with hashtags, we decided to use Instagram as the social network to perform the proposed system, because, for now, it's the platform where the users share content with more hashtags.

Instagram is a social network focused on sharing images and it's possible to use hashtags to link shared media to other publicly available content of the same topic or a similar one. Instagram is also known for not punish users and posts that use many hashtags, like many other Social Networks do. Instead Instagram limits the number of hashtags *per* post, up to a maximum of 30. This factor led users to use hashtags many hashtags *per* post, which is what we were looking for. This is the main factor we choose Instagram to follow the behaviour of hashtags.

First, the system needs to collect the data set that consists of the data of a pre-defined set of hashtags and information about the posts where they are used in and also their evolution through time.

With the data set complete the system can be trained to predict future values of popularity. The output of the implemented system is a graph with the curve that represents the future values of each hashtag.

The main goal of this thesis is therefore, to prove that it is possible to define the popularity of a hashtag and predict its values.

Chapter 2

State of the Art

In this chapter, a definition of popularity in social networks is explored. Previous work related with the problem of this project is also presented, as well as the state of the technology that is currently used to solve identical problems.

2.1 Popularity in social networks

Popularity or virality in social networks are hard to define, mainly because of the different metrics that are supported by each social network. For instance, Instagram posts can be liked or commented but have no negative metric. Other networks also have a similar system implemented, but with variations. Facebook has the popular like action, although in 2016 [Fac16] introduced *reactions*. Reactions are an extension of the Like button in order to allow people to express themselves in more ways and share their reaction in a community. Now, Facebook users can express if they get sad, amazed or angry with the content of a post, for example [Fac16]. Twitter also offers a like action, as well as the *re-tweet* that works in the same way as a Facebook's share.

Figure 2.1: Most *re-tweeted* tweet of all time

There are also networks that provide actions with negative implication, which is the case of Reddit that associates a score to each publication. That score is the amount of likes (called *upvotes*) minus the dislike values (called *downvotes*). Adding to this, the Reddit algorithm also takes in consideration how long the post has been created and calculates the relevance privately.

The like/dislike metric is a valid variable to take in account when dicussing popularity. However, it should not be the only one. It is usual to acknowledge something popular as something that is noticed by a large audience. That metric is known as *reach* and it is used by several advertisers and companies. For example, Youtube has a visualization counter per video that is normally used to measure that video's *reach*.

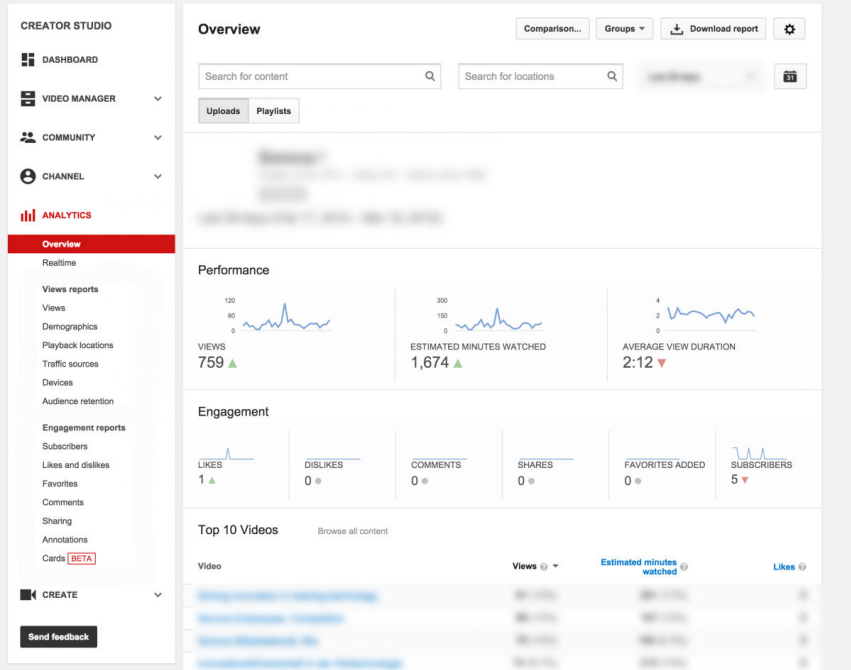


Figure 2.2: Youtube analytics page

The *reach* of a publication is not always revealed. On Facebook, and now on Instagram, there are subscriptions plans for companies or advertisers that provide these type of metrics. This happens because *reach* has a great advertising value. Other networks, like Youtube and Twitter, offer statistics that help their users analyse their publications' *reach*.

However, there isn't a platform that reveals the *reach* of a hashtag. Since hashtags work as pointers to the available information, knowing its *reach* is very important for an efficient use. There are already several platforms that try to solve this problem. All these platforms follow a business-to-business software-as-a-service strategy since it's where the big investment is but also offer some low cost plans, at the cost of less features, trying to reach smaller business or individuals.

Below are some examples of these types of services:

- Simply Measured [Sim]

- Keyhole [Key]
- Hashtracking [Has]

Since these platforms are all paid and closed-source, there's little information, if any, about the technology or even the algorithms that they use to achieve their statistics.

2.2 Similar work

2.2.1 Case studies

Several works attempt to understand how popularity behaves in social networks. Since Twitter was the first social network to feel the impact of hashtags, there are several case studies with this social network, such as:

[KMF+14] uses Twitter data to predict bursts and popularity of hashtags. To determine a burst of an hashtag, a window of 5 minutes is used to check if the count of a hashtag is greater than a threshold. This shows that the key metric that they use is the amount of times each hashtag appears in a time window. It's not the number of times that a hashtag is used that make a post more popular or not.

[WWL+16] is another interesting case study using Twitter. It analyses the data from two days during the Occupy Wall Street Movement in 2011. The authors identified popular hashtag types and examined hashtag co-occurrence patterns. It also provides a comparative analysis of how major types of viral hashtag may play different roles depending on different movement cycles. However, it suffers from the same mistake as the previous case study.

2.2.2 Tagzam

Tagzam is a mobile application used to automate the process of classifying photos with hashtags. It is focused in at social networks that support visual media, like Instagram, Facebook or Twitter. It receives as input a single image and analyses it. It returns two list of hashtags: one from image classification and another from the most used hashtags in social networks as shown in Figure 2.3.

After a quick analisys of the process used by Tagzam, there are some noticeable flaws.

First, the output hashtags of the image classifier are simple labels. These labels often are names of objects present in the image, which makes us believe that the background process for this task is using a image classifier service, like Microsoft's Computer Vision API. This API returns information about visual content found in an image. Information that includes tags, descriptions, and domain-specific fields as well as the confidence level that the model has on the prediction. Using a model of object recognition has one problem: it only identifies known objects. In a social network, hashtags can be phrases or made-up words that aren't in part of the dictionary. This type of hashtags is the one with greater value since they were created within a social network and become viral from the inside out.

(a) Recognized hashtags by Tagzam

(b) Popular hashtags by Tagzam

Figure 2.3: Tagzam predicted results for a cake photo

The second set of hashtags that Tagzam returns is a compilation of the most used hashtags in several social networks. This list offers no value because the hashtags in this set won't fit in the same context as the majority of photos used in Tagzam.

2.3 Multi-label image classification

The problem of classifying an image with a set of labels can be divided in two different problems: single-label classification and multi-label classification. Single-label classification is referred to the process of assigning a label to an image. About the multi-label classification, is assigns several labels to an image.

Single-label image classification has been largely studied in the recent past [FFFP07], [GHP07], [DDS+09]. With the evolution in the field of Neural Networks, one of the breakthroughs in image classification is the deep convolutional neural network (CNN) [KSH17]. It has the best performance in the large-scale single-label object recognition task, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [FFFP07] with more than one million images from 1,000 object categories.

Nevertheless, multi-label image classification is an even more complex problem and many methods [PSM10], [Yan12], [DXC+13] have been proposed. With the success of CNN on single-label image classification, a new motivation arouse but the CNN model cannot be extended to multi-label image classification [WXH+14]

Several methods have been proposed such as INRIA [HJS09] which is based on the transitional feature extraction-coding-pooling pipeline, CNN-SVM [SASC14] which directly applies SVM on the OverFeat features pre-trained on ImageNet, I-FT [WXH+14] that employs the squared loss function on the shared CNN features, HCP-1000C [WXH+14] which uses region proposal information to fine-tune the CNN features pre trained on ImageNet 1000 data set and CNN-RNN [WYM+16] that combines the advantages of the joint image and label embedding and label co-occurrence models by employing CNN and RNN (Recurrent Neural Network).

The proposed CNN-RNN method outperforms the I-FT method by a large margin, and it also performs better than HCP-1000C method, although the CNN-RNN method does not take the region proposal information into account. A comparison of the described methods on the PASCAL VOC 2007 data set is shown in [WYM+16].

Even though this thesis scope doesn't include image classification, the study about this matter was done with the intention of providing a theory basis for the future work.

2.4 Time series prediction

2.4.1 ARIMA - Autoregressive integrated moving average

Time-Series Prediction consists of the use of a model to predict future values based on previously observed values. This problem is typically applied to stock market [Kim03], [SPW98] where nowadays, more and more, data analysts are hired by top firms to build predictive models. This problem is also common in smart-systems where the behaviour of various subjects needs to be predicted, like traffic [TP15] or weather forecast [MYL+16], [MGYC16].

To solve this problem, the highly popularized Box-Jenkins autoregressive integrated moving average (ARIMA) model has been successfully applied in not only economic time series forecasting, but also as a promising tool for modeling the empirical dependencies between successive times between failures.

2.4.2 RNN - Recurrent Neural Networks

Regarding the increase of the popularity of Neural Networks, the motivation to apply this type of methods to this problem originated relevant breakthroughs. However, traditional neural networks don't support information persistence, which is a major shortcoming for Time Series Prediction. RNN address this issue. They are networks with loops in their structure, allowing information to persist.

In Figure 2.4, a unit receives X as input and outputs a value h. Since it has a loop, it allows the information to pass from one learning step to the next. In the last few years, there have

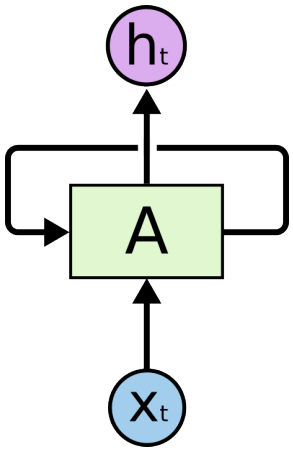


Figure 2.4: Recurrent Neural Networks have loops

been incredible success applying RNNs to a variety of problems: speech recognition, language modeling, translation, image capture and so on. One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame.

[HXG02] is a comparative study of the ARIMA model and two neural network architectures: multi-layer feed-forward network and RNN. Results showed that both ARIMA and the RNN outperformed the feed-forward network with lower predictive errors and higher percentage of correct reversal detection. When fine-tuning the feedback weights in the RNN, this method gave satisfactory performances compared to the ARIMA model.

2.4.3 LSTM - Long Short-Term Memory

LSTMs are a special kind of RNN that are capable of learning long-term dependencies. They were introduced by [HS97], and were refined and popularized by many people in [GSC99]. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

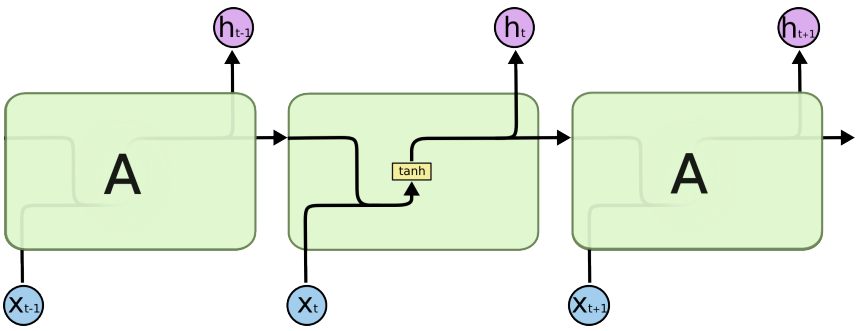


Figure 2.5: The repeating module in a standard RNN contains a single layer

LSTM also have a similar structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four.

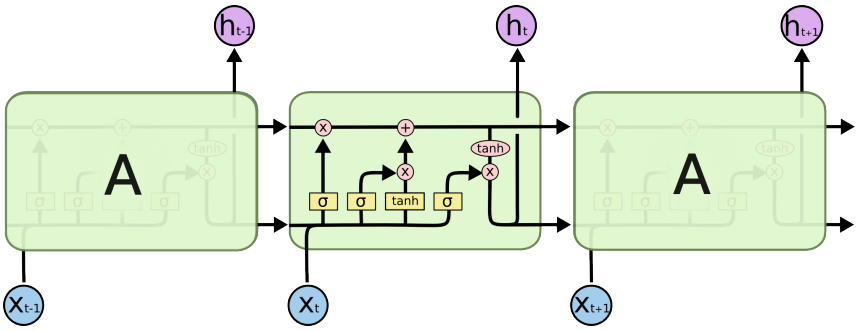


Figure 2.6: The repeating module in an LSTM contains four interacting layers

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. It runs along the entire chain, with only some minor interactions and it's very easy to just flow unchanged. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer (activation) and a pointwise multiplication operation. The sigmoid layer (σ) outputs numbers between zero and one, describing how much of each component should be let through. Zero means to not let anything go through while one means to let everything go through.

An LSTM has three of these gates, to protect and control the cell state.

LSTM has many use cases and solves several problems, like weather forecasting or stock market prediction. In [XNS15], a LSTM network is applied to model S&P 500 volatility, adding Google domestic trends as indicators of the public mood and macroeconomic factors. The trained model gives a mean absolute percentage error of 24.2%, outperforming other models.

2.4.4 Tools

Tensorflow was chosen for this work since it's is the most used platform and has more resources available. As of the writing of this thesis, Tensorflow has 87,318 stars on Github, the largest hosting service for OSS.

The second most used alternative is Theano, which has the same features as TensorFlow. However, Theano only has 7,700 stars on Github which means that not many people are using it comparatively to Tensorflow.

Tensorflow is the primary software tool of deep learning, created by Google. It is an open source artificial intelligence library that uses data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for:

- Classification
- Perception

- Understanding
- Discovering
- Prediction
- Creation

However, the learning curve of Tensorflow's API is steep. Because of the reduced time that this work had to be complete, we decided to use Keras [Ker].

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was designed with the developer experience in mind since it offers a simple API that minimizes the number of user actions required for common use cases. Also it provides clear and actionable feedback upon user error.

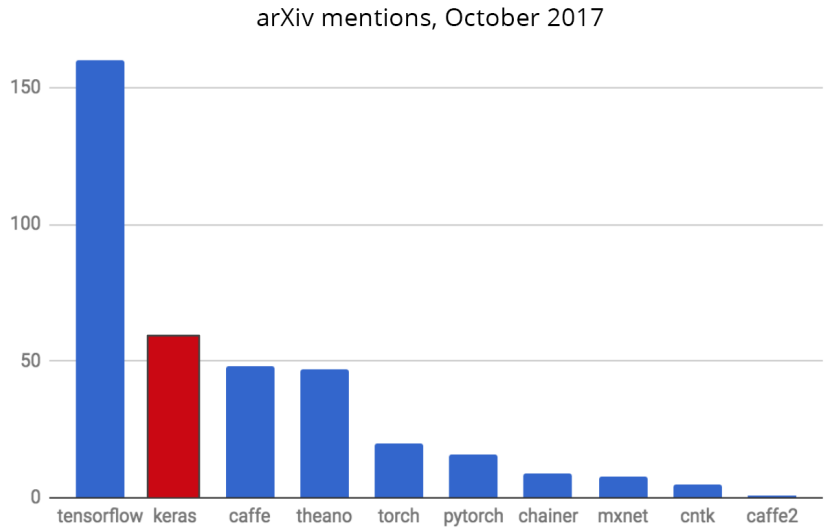


Figure 2.7: Keras is number 2 mentioned in scientific papers of arXiv

Keras was one of the most popular tools for Deep Learning amongst researchers, as shown in Figure 2.7, and amongst the OSS community with 24,680 stars in Github. And since it supports multiple backend engines, specifically TensorFlow, it was the choosen tool for this project.

Chapter 3

Data set

In this chapter a brief explanation of the Instagram API (IAPI) is given. It also highlights the problems faced to fetch information from Instagram as well as the requirements of the data set. In the end of this chapter we will present the adopted solution to meet the requirements.

3.1 Instagram API

The official method to fetch data from Instagram is to use its API Platform, which is divided in two modes: **sandbox** and **live**. The sandbox is for new apps or beginner developers, whereas the live mode provides access to full data of Instagram.

3.1.1 Sandbox mode

Sandbox mode is the kick off point for every app that uses Instagram API. It simulates a fully functional yet private environment that allows developers that are new to Instagram to explore its fetures. It's also useful for teams that need multiple clients for development, staging and other non-live environments.

Since this mode creates a detached environment for each registered app, it has some limitations:

- Restricted to 10 users.
- Data is restricted to the 20 most recent media from each user.
- Reduced API rate limits.

The limitations of this mode make it only useful for proof of concept applications in a very restricted scale. In order to leave sandbox and use the live mode, it's required to submit the app for review.

3.1.2 Live mode

In **Live mode** it is possible to have full access to Instagram's live data. It exposes an interface to be able to fetch public data that's available in Instagram database. This is the right mode to use if the application is to become public and hit a medium-large scale user base.

The app has to be approved in order to go live and to have fully access to Instagram content. Nevertheless, even if the review is positive, it's not guaranteed that the app has granted permission for every scope of the IAPI.

IAPI use cases

- To install a third party widget to show Instagram content on my website;
- To display hashtag content and public content on my website;
- To display my Instagram posts on my website;
- To build analytics for my own Instagram account;
- The app is still in development and/or is a test app;
- To allow non-business users to login and post comments, likes or follow actions;
- To allow people to login with Instagram and share their own content;
- A product that helps brands and advertisers understand, manage their audience and media rights.

The request is reviewed and the app is re-evaluated to check if it fits into an use case that needs permission. The following items must be fulfilled for a permission review:

Submission quality - All notes must be clear, concise and in well-written english;

Video screencast - A clear and working screen-cast of the app must be provided displaying the login experience, proper credentials and the usage of every permission requested. For new apps, this video must be done using **sandbox** mode;

Development Phase - Only production versions are reviewed (ready to be live);

Policy compliance - The developer needs to agree with Instagram Platform Policy and Instagram Brand Guidelines.

Due to the limitations and restrictions of the IAPI, it was not a valid choice to fetch the required information. On one hand, there's no match between a use case and the context of this thesis. On the other hand, even if there was a suitable use case, the requirements for the submission needed a proof of concept using **sandbox mode**. Since the proposed system could only work with a large set of real data, this mode wasn't suitable.

Due to the mentioned reasons, we had to look for other options than IAPI. Since Instagram has a web application capable of getting the required public information, the best option was web scraping.

3.2 Web scraping

In order to replace the use of IAPI, we decided to use web scraping to build the data set. To use the purposed system, the web scraper had to perform some fundamental tasks. Most of these tasks were about fecthing data from Instagram web application.

Web scraper requirements

- Fetch information about a hashtag;
- Fetch the top posts of a hashtag;
- Fetch the most recent posts of a hashtag;
- Fetch information about a post by its identifier;
- Fetch likes and comments of a post by its identifier;
- Fetch information about a user;
- Fetch the most recent posts of a user;
- Fetch a user's *username* by its identifier;

With this list of requirements, we proceed to research what existing tools could perform this set of functions.

Several Open-Source project were found, but only four projects were analysed: [Ily16], [Gat17], [Arc17] and [Pos17].

[Ily16], [Gat17] had the last commits dated of over an year ago which meant that they weren't updated. Since we are dealing with web page scraping, there are frequent updates that make these options not viable.

The other two [Arc17], [Pos17] were popular and maintained with frequent commits to the source code. However, both solutions opt to use hardcoded values in order to query Instagram. This is fine in an environment where we can stop the system to update. Yet, the aimed system was meant to work in an environment without downtime. The system must follow the likes' evolution of a bulk of posts and if the data that is being analized didn't fully followed its evolution, it should be considered invalid. Due to this reason, this option was eliminated.

At this point, we decided that it was best to build a web scraper to better suit the system's needs. We already had set a list of requirements and needed to choose the technologic stack to use in this project.

Added to the list of requirements, this scraper had to overcome the fault that [Arc17], [Pos17] had and instead fetch dynamically the Instagram web-application for the query values so it could be used for long periods of time. To fetch the query values, a headless browser was used since it recreates a web browser without a graphical user interface. This option provided automated control of Instagram's web application in an environment similar to a real browser. Since Chrome is known for its compatibility and stability, and had released a Headless mode recently, it was the strongest option.

Headless Chrome provides, along with a command line interface, a fully featured high-level API. The Headless Chrome API is written in Javascript and is called Puppeteer [Aut17]. Since it is written in Javascript and scraping consumes a lot of web objects, the stack chosen to develop this web scraper was Javascript, since it has many useful modules and function to deal with web requests and responses. This module, the web scraper, was heavily inspired by Instagram's API so its endpoints are recreations of the official API.

The developed scraper API has three modules: **Media**, **Tags** and **Users**. The other modules provided by the Instagram API, like **Relationships** and **Locations**, were left out because there was no use case for them in the proposed system and the modules **Comments** and **Likes** were integrated into the three implemented modules. Also to have in consideration, only the endpoints related to fetching information were implemented since the goal was to build a data set and not to publish content to Instagram.

3.2.1 Web Scraper API

This subsection functionally documents every endpoint developed API for the Web Scraper showing its module architecture.

Media Module

Get Endpoint to get information about a media object.

Receives *shortcode* as a parameter, which is used to identify the media object. Even though there's also an ID, the shortcode is the one used for queries.

Comments Endpoint to get a list of comments on a media object.

Receives *shortcode* and *limit* as parameters, so the *shortcode* used to identify the media object and *limit* to represent the number of (most recent) comments to be returned.

Tags Module

Info Endpoint to get information about a tag object.

Receives *hashtag* as a parameter, which is used to search for the tag object.

Recent Endpoint to get a list of recently tagged media.

Receives *hashtag* and *limit* as parameters, being *hashtag* used to identify the tag object and *limit* to represent the number of media objects to be returned (by most recent).

Top Endpoint to get a list of the most popular tagged media.

Receives *hashtag* as parameter which is used to identify the tag object and returns nine media objects.

Search Endpoint to get a list of tag objects that result of a string query using the parameter *hashtag* in order of similarity.

User Module

Info Endpoint to get information about a user object.

Receives *username* as a parameter, which is used to identify the user.

Posts Endpoint to get a list of the media objects published by a user.

Receives *username* and *limit* as parameters, being *username* used to identify the user and *limit*, an optional parameter, to represent the number of media objects to be returned (by most recent).

Username Endpoint to get username of an user by its ID.

Receives *ID* as parameter which is another identifier for an user, although never used as parameter in other endpoints, many media objects refer only to this field to identify its owner.

In order to assure that the web scraper was still up-to-date with the Instagram web application, several integration tests were developed. They tested every endpoint and several cases of different parameters and results. The tests used full implementation and environment, querying the Instagram's web application and comparing values with the expected ones. This process allowed to identify some changes to the Instagram web application and quickly change the specific endpoint to accommodate those changes.

Due to the utility of the web scraper, the same was published as OSS in Github, receiving several comments identifying issues and opening the discussion about the implemented interface and architecture.

It was also published on *npm* (Node Package Manager), the Javascript's most popular package manager which enabled many other developers to integrate this web scraper in their projects.

3.3 Selection of a set of hashtags

As said before, the proposed system is only a proof of concept. Therefore, the data set can't self-expand as the complete system described in 1.3. Instead, only a small sample of hashtags could be considerated to be analysed through time. The selection process couldn't be random to assure that

each selected hashtag were significant enough. However, the set couldn't also be too generalist or abstract, where many types of posts could relate with that hashtag.

In order to meet the requirements of the data set, we performed an automated graph search. Since the data was saved in graph form, neither a relational database nor a NoSQL database was appropriate for this task. Because of that, a database with a dedicated graph engine was a better fit. Considering the options of graph databases, Neo4J was the most established, providing more resources to learn and debug. It also had a Javascript and Python driver, which was useful to integrate with the web scraper and the analysis scripts (written in Python).

The graph search started with a single hashtag (seed), from which the Instagram's top posts were fetch, which always returns nine posts. From those posts, any hashtag found in the posts caption and/or comments, would serve as a new seed.

Any new hashtag found was added as a vertex in the graph and the hashtags used in the same post, were added as edge between the two hashtags' vertices. Any hashtag that already existed in the graph increased the weight of that vertex and added new or increased the edges weights.

This graph search started with hashtag *dogwood* as initial seed and ran for exactly 30 minutes resulting in 17599 hashtags (vertices) found and 41777 relations between hashtags (edges). *dogwood* was choosen as the initial seed because it's used in as a self-challenge for photographers and every week there's a new challenge, with a different theme. For each challenge, the hashtag *dogwood* is used, as well as the hashtag *dogwoodX* where *X* is the number of the challenge's week. Since *dogwood* is often used with its week-specific hashtags, it was considered a good option for an initial seed providing the graph search a good variety of hashtags.

The graph search returned much more hashtags than those we could use for the data set, so a process of selection was made. Since new hashtags were discovered based on the relations to the known hashtags, many had similar semantic value and could be aggregated in clusters.

To divide the data into clusters, we used the walktrap algorithm [PL06]. Walktrap tries to find densely connected subgraphs via random walks. The idea is that short random walks tend to stay in the same cluster.

With the data aggregated in clusters, we could then proceed to specify which metric would decide what hashtags were the best to use.

In order to solve this problem, we used Google's Pagerank [PBMW99] algorithm to assign to each hashtag a rank, in the global context (not within its cluster). Pagerank was proposed to estimate the probability of browsing to each site on the internet based partially by the number of other Web pages that link to the target page. However, this works for any kind of graph, it computes a kind of *centrality* that measures how well connected each vertex is to other well connected vertices.

Having the hashtags ranked and grouped in clusters, made it possible to select which ones were more relevant and without repeating many hashtags with the same semantic value. To complete this process, only the clusters with more than 10 hashtags and less than 20 hashtags were considered. These limits seemed fair values for what was considered a cluster with very abstract

hashtags (clusters with more than 20 hashtags) and a very specific cluster in which its hashtags only had a narrow meaning (clusters with less that 10 hashtags).

In the end, the three better ranked hashtags of each resulting cluster, were picked and chosen as the ones to analyse, resulting in 106 hashtags. An example of some hashtags' clusters in shown in Appendix A.1.

3.4 Data over time

Considering the set of hashtags previously defined, the system had to continuously analyse them. This process was divided into three separate jobs: *Collector*, *Watcher* and *Analyst*.

The *Collector* job was responsible for fetching the 6 most recent posts that used any of the selected hashtags. Then it would invalidate any post that was not published in the last hour. Also, posts from users without at least 50 posts were considered invalid. Lastly, it would save which posts were valid.

The *Watcher* job was only responsible for fetching the amount of likes for each active post and save it. A post is considerated active if it is still in its lifetime. A post reaches the end of its lifetime when the amount of likes hasn't changed in the past 6 read-outs.

The task of comparing the past results and deactivating it, is the responsability of the *Analyst*. The complete architecture is illustrated in Figure 3.1.

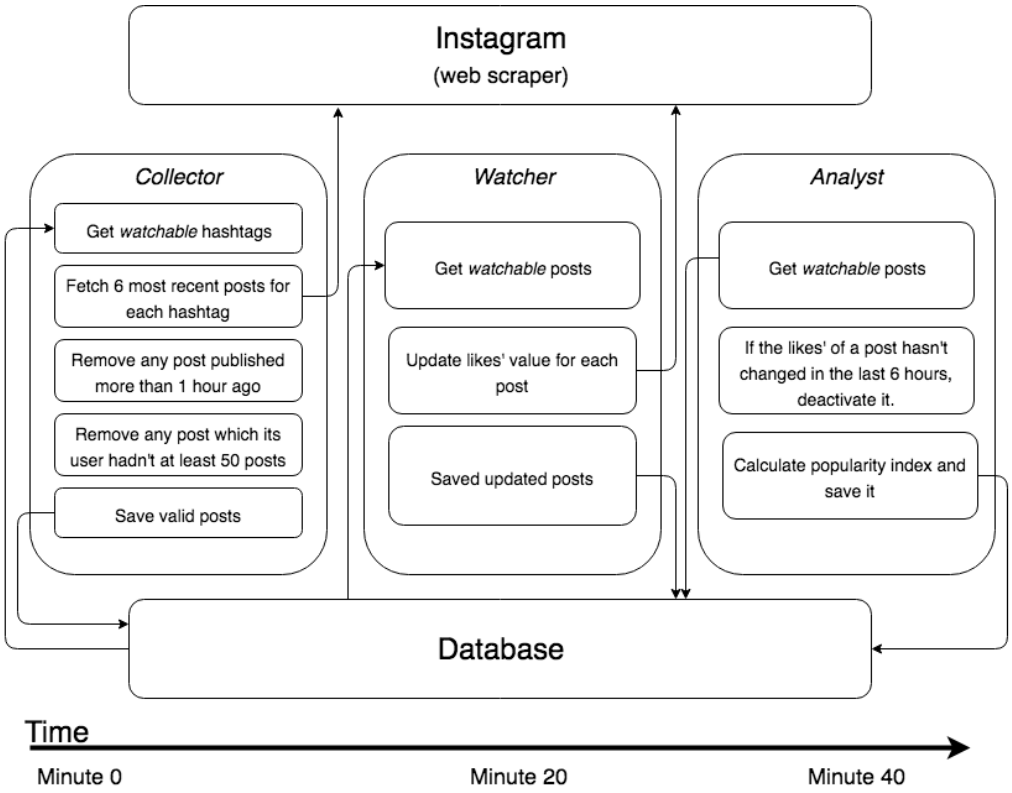


Figure 3.1: Dataset over time architecture

These jobs were triggered by *crontab*, which is a time-based job scheduler in Unix-like computer operating systems. Is is often used to set up and maintain software environments that need to schedule jobs to run periodically at fixed times, dates, or intervals. Each of the specified jobs would run once in each hour, starting with *Collector* at minute 0, then *Watcher* at minute 20 and finally *Analyst* at minute 40. This division assure that the previous jobs ran before the current job started, with the amount of hashtags considered.

This collection of the data set started at December, 29 of 2017 and ended in January, 19 of 2018, resulting in a total of 3 weeks of data.

Chapter 4

Popularity Prediction

In this chapter what the popularity index means and how it is calculated is discussed. The process of predicting future values of a hashtag's popularity index based on the previous values is also described.

4.1 Popularity Index

Popularity index is a value that determined to be representative of a hashtag's popularity effect on a social media post. It can either take positive or negative values, depending on the effect that it has. The zero value means that it has a neutral effect, and the use of that hashtag is neither positive or negative.

An inefficient use of a hashtag, can have negative effect due to maximum number of hashtags that Instagram imposes. Remember that the goal of this work is to find the most efficient use a hashtag can have.

As mentioned in section 3.4, when a post X reaches the end of its lifetime, the *Analyst* flags X. At this point, the function that represents the linear regression of the 50 post's likes, of the same owner, prior to post X is calculated.

The aforementioned function is used to calculate the expected value of likes for X at the end of its lifetime. The expected value is subtracted to the real value of likes obtained at the end of X's lifetime.

$$index_X = \frac{real_X - expected_X}{m_2} \tag{4.1}$$

In the end, the result is divided by the sample variance (m_2) of the same 50 posts prior to X. The sample variance is the sum of squared deviations from the mean. It is distinguished from the variance by dividing by the length minus one, instead of dividing the sum of squared deviations by the length of the input. This corrects the bias in estimating a value from a set that isn't fully known. Therefore, the sample variance is used as a normalizing factor between users.

This was required since, for many users, a variation of 500 or 1000 likes doesn't represent a meaninful variation. While to others, more 10 or 20 likes is a big variation.

The result of the equation 4.1 is the popularity index from post X. This value is pushed to an array of indexes for each hashtag used in X. It's possible to get which hashtags were used by searching X's caption and comments with the regular expression: `#\w*`.

4.2 Predicting future values

Having a list of popularity indexes through time for each hashtag allows us predict new values as explained in Section 2.4.

The model used in this work was inspired in [Rav16] and [F16]. Both models use a LSTM network to forecast stock prices using price history and follow the same architecture of layers:

LSTM Default LSTM layer. Returns one output for each input time step and provides a 3D array. This approach is needed when stacking LSTM layers.

Dropout Consists of randomly setting a fraction rate of input units to 0 at each update during training time, which helps prevent overfitting.

LSTM Default LSTM layer. It doesn't returns output since it's the last LSTM layer for this model.

Dropout Used with the same intention as the other Dropout layer, to prevent overfitting.

Dense Regular layer of neurons in a neural network.

Each neuron receives input from all the neurons in the previous layer. It's used as an intermediate activation layer.

Dense Another regular layer of neurons in a neural network. This last one defines the final output of the model.

The models differ when choosing the network optimizer. [Rav16] chooses RMSProp optimizer [HSS] which divides the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight. [F16] chooses Adam optimizer [KB14], an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.

Both approaches used MSE to compile the loss value of each model.

The data used to feed the model was relative to the three weeks of data. Since the goal was to create a single model that could predict popularity indexes for any given hashtag, the same model would needed to be trained with data from any given hashtag.

Regarding this, all the training data must all be the same size. Since many hashtags did not have enough popularity indexes to train a significant model, reducing every row of data to the minimum length would mean a significant loss in accuracy and waste of data. Therefore we decided to only look at the hashtags with at least 70 values for popularity index. This process left 25 hashtags with 70 popularity indexes.

The data was fed into the model in sequences, since the problem is to predict the next values of the time series. The row of data was split into a set of sequences, rather than a larger sequence. The sequences had a fixed length, named *window*, which is considered a hyperparameter.

Hyperparameters express "higher-level" properties of the model such as its complexity or how fast it should learn. They are usually fixed before the actual training process begins. Each hyperparameter was subject to a set of experiments to better understand how they behave for the given data set.

Other hyperparameters considered were the number of *epochs* and the *batch size* of the network. These two hyperparameters go hand in hand.

Epochs represent the amount of times a network go through the training set. The *batch size* defines the number of samples that are going to be propagated through the network. The model is updated each time a batch is processed, which means that it can be updated multiple times during one epoch. If *batch size* is set equal to the length of the input, then the model will be updated once per epoch.

In order to test the model, the data was split into two parts: 90% to train and the last 10% to test. After training, the last 10% were used to compare the predicted results with the real ones and calculate the MSE with the following formula:

$$MSE = \frac{1}{n}\sum_{i-1}^{n}(predicted_i - real_i)^2 \tag{4.2}$$

This, however, gave us an MSE value of the model for each hashtag. We needed a metric that could evaluate the whole model in order to be possible to test different hyperparameters and achieve its best result. The mean of MSEs was therefore used, as we refer as MMSE.

A series of experiments were made exploring the different combinations of hyperparameters.

		Window		
	MMSE	3	4	5
Epochs	100	0,06200	0,05884	0,05945
	500	0,05804	0,05791	0,06123
	1000	0,09935	0,06309	0,06480

Table 4.1: MMSE for *batch size* 32

First, we fixed the *batch size* in 32, because it was the default and recommended size in Keras. In Table 4.2 the results of the same tests are shown, but with *batch size* of 16.

The *window* hyperparameter was rotated between 3 and 5. Lower than 3 meant that the sequences to train had length of 2 or less which wasn't considered a significative sequence. Higher than 5 was not used since the MMSE values didn't decreased over the increase of the *window* size.

The number of *epochs* was also tested. Changing the number of *epochs* gave us feedback if the network was overfitting or not. Since the work in which our model was inspired used 300 epochs, we decided to conduct the experiments using 100, 500 and 1000.

		Window		
	MMSE	3	4	5
Epochs	100	0,05488	0,06304	0,06221
	500	0,07425	0,05953	0,06415
	1000	0,06788	0,06287	0,05651

Table 4.2: MMSE for *batch size* 16

Even though we used Dropout layers to prevent overfitting, it was noticeable the tests that used 1000 *epochs* never achieved the lowest MMSE. This means that this value of *epochs* is too big for our data set. The values of the tests that used 100 and 500 *epochs* weren't as conclusive as the test with 1000 *epochs*. Considering this, we can conclude that that in these tests the did not overfit and the other hyperparameters were the ones relevant to the changes in MMSE.

With this set of experiments, the best set of hyperparameters for our data was found, as shown in Table 4.3.

Hyperparameter	Value
Batch size	16
Epochs	100
Window	3

Table 4.3: Optimized hyperparameters

4.3 Results

The low MSE, seen in Table 4.1 and 4.2, were indicative that the predicted values were close to the real ones.

Looking at the graph for the predicted values, like in Figure 4.1, these values follow the tendency of the actual data. However, when the actual data graph quickly varies, the model doesn't do well. This is due to the fact that model was trained with the data of every hashtag, so it learns the more regular patterns. When the actual data misses a known pattern of the model, this one fails.

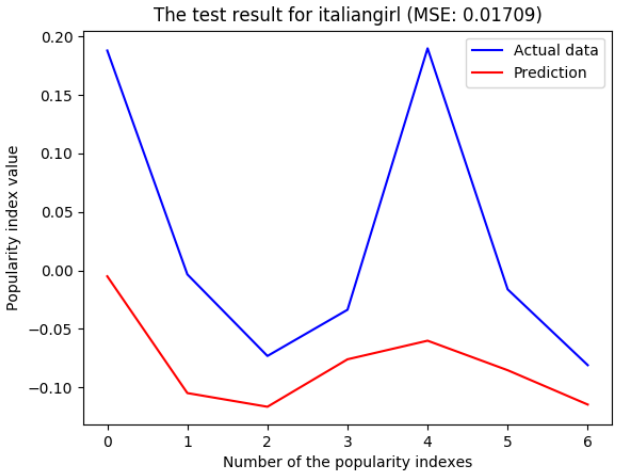


Figure 4.1: Predicted results for *#italiangirl*

This is even more noticeable in the results for *#swimsuit*, Figure 4.2, which is the hashtag with the higher variation in popularity index and consequently the one with higher MSE.

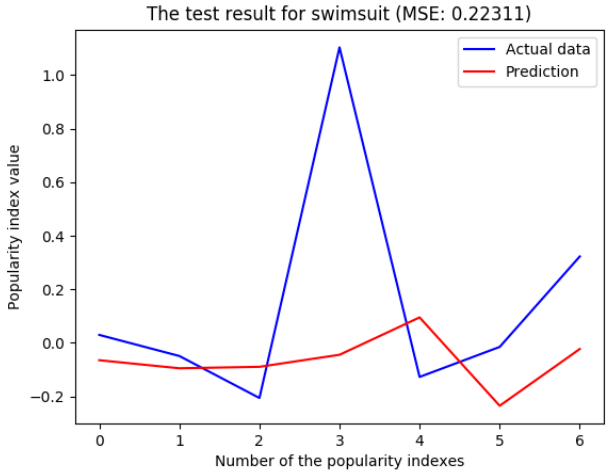
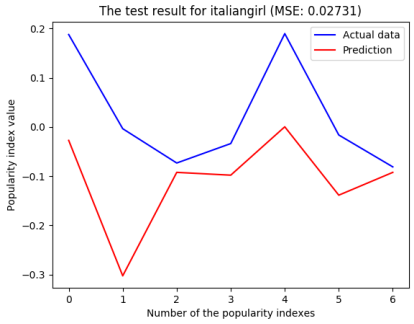
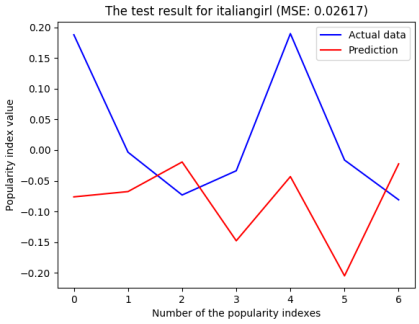


Figure 4.2: Predicted results for *#swimsuit*

We also compared the same hashtag with the different models described in Section 4.2 where we tested different hyperparameters. In Figure 4.3 it's possible to see that the model does not fit well to the data since it can't recognize the correct pattern for this hashtag. In Appendix A.2 it is shown all the results for the different hyperparameters.



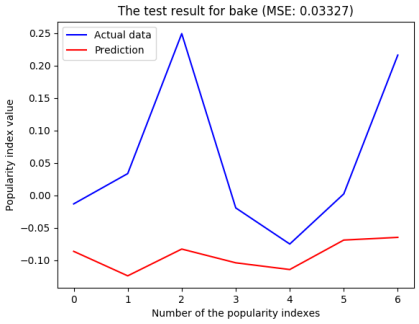
(a) Predicted results with 500 epochs



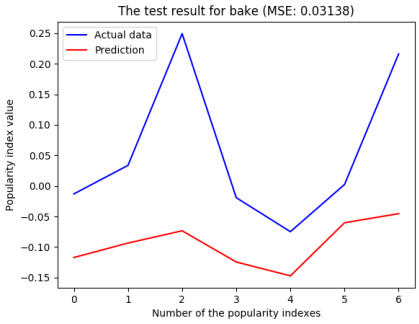
(b) Predicted results with 1000 epochs

Figure 4.3: Predicted results for *#italiangirl* with *window* of 3 and *batch size* of 16

Using the predictions that resulted, in Figure 4.4, from the model with different *batch sizes* we can verify that the results are very similar.



(a) Predicted results with *batch size* of 16



(b) Predicted results with *batch size* of 32

Figure 4.4: Predicted results for *#bake* with different *batch sizes*

These results show that we are able to predict the near future behaviour of a hashtag popularity effect. This popularity effect is reflected onto the posts that use that hashtag and it represent the exposure boost that the hashtag “gives" to a post.

Understanding the peak relevance of a hashtag allows users to apply that hashtag efficiently. This allows users to schedule their publications accordingly to the popularity of each desirable hashtag or, choose what hashtags are better options in a given moment. Having this information is particularly important for users that already have media content and do not know what hashtags will provide more exposure.

This tool opens the possibility for users to have a bigger influence in Social Networks.

Chapter 5

Conclusions and future work

5.1 Conclusions

This work had two major tasks:

- Create the means to build a Data set
- Create a Neural Network model that was able to predict a hashtag's popularity index

To create the data set and since Instagram's API wasn't a viable option, we developed a web scraper that used Instagram Web Application to fetch data. That tool was later published as OSS in Github and on NPM.

We decided to limit the data set to a predefined set of hashtags to reduce the computational need. The data set was then limited to a pre-defined set of hashtags that resulted in a graph search that was then grouped into clusters and picked the hashtags with higher Pagerank score.

The data set was complete with 3 weeks of data.

A Neural Network was then implemented using the LSTM architecture. Several experiments were conducted to find the best hyperparameters for the current data set. The final hyperparameters resulted in a Mean of each hashtag prediction's MSE of **0,05488**.

5.2 Future Work

5.2.1 Continuous learning system

Regarding future work, to make the system complete as it was conceived and described in Section 1.3, we would start by making the system autonomous. For it to be autonomous, it would have to self increase its data set.

One approach could be to have a continuous graph search as the one explained in Section 3.3. This graph search will keep finding new hashtags and adding them to the database. This process is the easiest option since it doesn't require aditional changes to the code of this work. The only concern with this option is the amount of load that the *Collector* and *Watcher* would have, since

these processes perform asynchronous tasks related to fetch information and have a limited time to complete their tasks.

Out of curiosity, we decided to save the hashtags found when a post reached the end of its lifetime. Those hashtags were added to the database but were flagged to not be analysed. However, by the end of the 3 weeks of data, the database had a total of 32812 hashtags found. This makes us believe that starting with a small number of hashtags and letting the *Analyst* find new hashtags is a viable option to grow the data set. This option requires only small changes to the current code of this work and has the same concerns as the first option.

5.2.2 Content analysis

The content analysis component would be the next part to implement.

The association of the content with hashtags is another feature we would like to improve we wish to prove, specially in case of image classification. Being able to automatically assign hashtags to an image is something that hasn't been done before. This is a hard task since it is not a simple classification problem; hashtags can be can be abstract and its posts have images with different meanings and totally different content.

5.2.3 Bi-directional LSTM

Although the prediction results of the time series are satisfying, trying different Neural Networks architectures would be beneficial to understand which model performs better for the data set of this project.

Specially, Bidirectional RNNs which are based in the idea that the output at a time may not only depend on the previous elements in the sequence, but also future elements. For example, to predict a missing word in a sequence you want to look at both the left and the right context. Essentially, these are just two RNNs stacked on top of each other. The output is then computed based on the hidden state of both RNNs.

5.2.4 More data

Due to the time restritions of this work, the maximum amount of data that we could gather was 3 weeks.

We believe that a bigger interval of time would give enough data in order for the model to be more accurate and to better predict the points of higher oscillation. With more data, the model would be better trained.

Having more data is also related with the system's feature of being autonomous, explained in Subsection 5.2.1. This is because, with the system being autonomous, it would also have to train itself regularly without human intervention.

References

[Arc17] Richard Arcega. Instagram Scraper, 2017.

[Aut17] The Chromium Authors. Puppeteer, 2017.

[BB11] Axel Bruns and Jean E Burgess. The use of twitter hashtags in the formation of ad hoc publics. In *Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011*, 2011.

[DDS+09] Jia Deng, Wei Dong, R Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database, 2009.

[DXC+13] Jian Dong, Wei Xia, Qiang Chen, Jianshi Feng, Zhongyang Huang, and Shuicheng Yan. Subcategory-Aware Object Classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[F16] Ben F. Neural Network with Financial Time Series Data, 2016.

[Fac16] Facebook. Using Facebook Brand Assets, 2016.

[FFFP07] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, apr 2007.

[Gat17] Thibaut Gatouillat. Instagram Scraper, 2017.

[GHP07] Gregory Griffin, Alex Holub, and Pietro Perone. Caltech-256 Object Category Dataset. Technical report, 2007.

[GSC99] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[Has] Hashtracking.

[HJS09] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 237–244. IEEE, sep 2009.

[How08] Bill Howard. Analyzing Online Social Networks. *Communications of the ACM*, page 3, 2008.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSS] Geoffrey E Hinton, Nitish Srivastava, and Kevin Swersky. Overview of mini-batch gradient descent.

[HXG02] S L Ho, M Xie, and T N Goh. A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computers and Industrial Engineering*, 42(2-4):371–375, 2002.

[Ily16] Ilyapt. insta-scraper, 2016.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Ker] Keras. Keras.

[Key] Keyhole.

[Kim03] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, sep 2003.

[KMF+14] S Kong, Q Mei, L Feng, F Ye, and Z Zhao. Predicting bursts and popularity of Hashtags in real-time. In *SIGIR 2014 - Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 927–930, 2014.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, 60(6):84–90, 2017.

[Mav17] Mavrck. 2017 Facebook User-Generated Content (UGC) Benchmark Report. Technical report, Mavrck, 2017.

[Mes07] Chris Messina. Groups for twitter; or a proposal for twitter tag channels. *Personal Blog: FactoryCity: http://factoryjoe. com/blog*, 2007.

[MGYC16] Anbo Meng, Jiafei Ge, Hao Yin, and Sizhe Chen. Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. *Energy Conversion and Management*, 114:75–88, apr 2016.

[MYL+16] Zhongxian Men, Eugene Yee, Fue-Sang Lien, Deyong Wen, and Yongsheng Chen. Short-term wind speed and power forecasting using an ensemble of mixture density neural networks. *Renewable Energy*, 87:203–211, mar 2016.

[NH13] Karine Nahon and Jeff Hemsley. *Going Viral*. Wiley, 2013.

[OW15] Jonathan A. Obar and Steve Wildman. Social media definition and the governance challenge: An introduction to the special issue, oct 2015.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[PL06] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218, 2006.

[Pos17] Postaddict.me. Instagram PHP Scrapper, 2017.

REFERENCES

[PSM10] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 143–156. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[Rav16] Siraj Raval. How to Predict Stock Prices Easily, 2016.

[SASC14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2014.

[Sim] Simply Measured.

[SPW98] E.W. Saad, D.V. Prokhorov, and D.C. Wunsch. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9(6):1456–1470, 1998.

[Ste17] Michael Stelzner. 2017 Social Media Marketing Industry Report. Technical report, Social Media Examiner, 2017.

[TP15] Yongxue Tian and Li Pan. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 153–158. IEEE, dec 2015.

[WWL+16] Rong Wang, Rong Wang, Wenlin Liu, Wenlin Liu, Shuyang Gao, and Shuyang Gao. Hashtags and information virality in networked social movement: Examining hashtag co-occurrence patterns. *Online Information Review*, 40(7):850–866, 2016.

[WXH+14] Yunchao Wei, Wei Xia, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. CNN: Single-label to Multi-label. jun 2014.

[WYM+16] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. CNN-RNN: A Unified Framework for Multi-label Image Classification. apr 2016.

[XNS15] Ruoxuan Xiong, Eric P Nichols, and Yuan Shen. Deep learning stock volatility with google domestic trends. *arXiv preprint arXiv:1512.04916*, 2015.

[Yan12] Shuicheng Yan. Hierarchical Matching with Side Information for Image Classification. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3426–3433, Washington, DC, USA, 2012. IEEE Computer Society.

REFERENCES

Appendix A

Clusters

A.1 Example of clusters

greecelover_gr, tv_greece, abandoned_gr, igs_world, expression_greece, streetart_addiction, loves_mediterraneo, pict_lovers, loves_athens, hubs_united, hdr_greece, loves_united_islands, igworldclub_hdri, loves_greece_, stunning_greece, visual_heaven, great_street_photos

---

stayandwander, bestvacations, beautifuldestinations, liveoutdoors, reiseblog, reisen, travelblog, weroamgermany, mountainlovers, fantasticearth, mountain, breathtaking, berlingermany, alpspitzkick, alpspix, alpspitzbahn, soulfulmoments, sinnigetiefe, kodakmoments, landscapelover, pocket_germany, bildrauschen79, naturebrilliance, essen, german, currywurst, alexliebtdiewelt, berliner, berlinstyle, instaberlin, welovecitywest, wheninberlintours, meindeutschland, tagsforlikes, citylife, city, deutschland, brandenburgtor, diestadtberlin, igersberlin, berlinstagram, berline, ehrenmann, schlumpf, stadt, coberlinmoment, ickbinneberlinerin, ichbineinberliner, iloveberlin, lenipepunktontour, familientag, rva, berlin, explorer, happythanksgiving, fresh, local, carbload, germanfood, thesix, toronto, eat, ig_berlin, visit_berlin, berlinbreeze, ig_deutschland, diewocheaufinstagram

---

seeaustralia, visualambassadors, rashventures, thisisqueensland, tree_captures, cruising_australia, bns_sky, skymasters_family, sunrise, longexposure_shots, long_exposure, exeptional_pictures, forbestravelguide, globeshotz, thevisualcollective, folkscenery, longexpo_addiction, mybrisbane, ourlonelyplanet, gottolove_this, theimaged, canonaustralia, shipwreckphotography, longexpoelite, byronbay, discover_earthpix, earthofficial, nisifiltersanz, openmyworld, earthvacations, love_united_trees, bns_tree, nudgee, adelaide, sunset_madness_, fs_sunset, bomboquarry, canonphotography

---

wewantbeer, beerhappy, growler, beerpub, trainingday, thebosshoss, babyatabrewery, russellbeerpub, stonycreekbrewery, prohibition

---

dubaibay, babydubai, beverlyhills, mydubai, newmother, dubaifood, kuwaitfashion, dubaimothers, dubaifashion, mariamhussein, actresslife, actress, kidsdubai

---

detailoftheday, detailfashion, outfitblogger, sweater, detailpost, dailylook, fromwhereistand, fashioninsta, detailoutfit, currentlywearing, fashiondiarys, detaillover, dailyoutfit, whatiworetoday

---

instablog, motheranddaughter, mothersday, mamablogger, mamablog, motherhood, baby, kids, liberty, melbournelife, niceday, modernlayette, gifts, monicaandandy, giftsforher, mother, makelovetothemirror, sheesh, guccibelt, gotmelike, rain, nawlfr, gucci, motherson, motherdaughterlove, toptags, mothers, motherdaughter, mothermonster, motherandson, parenthood, motherslove, bestmomintheworld, mommylove, instamom, motherdaughtertime, mothernature, mothersday2017, soblessed, motherlove, blogimam, mothercare, mothership, babies, roundwegowithsuedaley, roundwegosal, mimikids7, kidstagram, instagram_kids,

kidsphotography, hipstamama, hipstamama_nsk_kids, hipstamama_nsk, instakids, lovemommy, lawwdhavemercy, coolerbeitrag

---

flowerlovers, flowermedicine, flowerinbox, greenwitch, calibouquet, herbalmedicine, herbalism, plantperson, planthealing, plantmedicine, floweressences, heartchakra, lovetheearth, redflowers, australiannativeflowers, bushflowers, plantlovers, sacredplants, shamanichealing, goddessrisingsisterhood, antiinflammatory, organicgardening, botany, plantknowledge, healthydigestion, pachamama, goddesskali, healthyheart

---

freepattern, sewsweetnesspattern, kismettrinketboxes, rainbow, andoverfabrics, alisonglass, agchroma, annexdoublezipboxpouch, easytosew, oslocraftbag, greatgiftidea

---

louisvuitton, vernis, bagmania, pommedamour, makeupbag, makeupbrushes, michaelkorshandbag, lv, lvoe, lvaddict, dolcegabbana, marcjacobs, loveformichaelkors, michaelkors

---

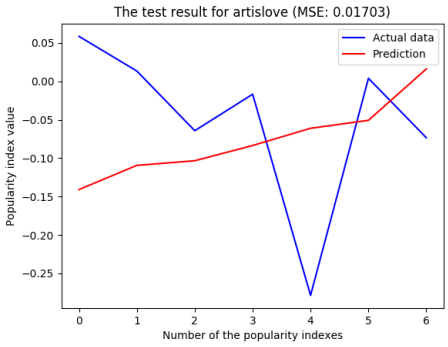
vintage_shotz, total_rural, the_detail_lens, portasejanelas, rainbow_wall, tv_retro, rustlord, arte_of_nature, pocket_colors, tv_hiddenbeauty, wallfilth, tv_doorsandwindows, jj_beautyofrust, lovers_home4, total_colors, rottenfeed, germandecay, patina_perfection, thehub_vsco, jj_texture, infinity_rust, jj_doorsandwindows, vintagedecor, icu_doorsandwindows, rust_of_our_world, thehub_details, antique_r_us, portaseportoes, flaming_rust, be_one_doorsandwindows

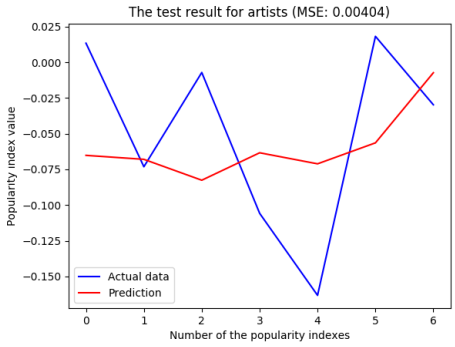
---

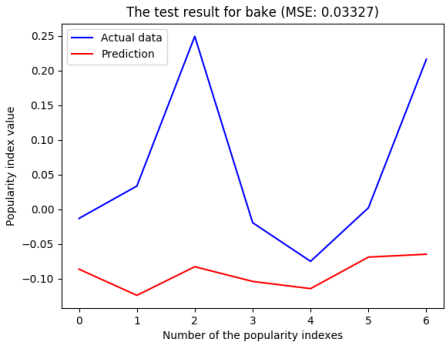
lovemyjob, favoritejob, mypassion, singing, mylife, favoritework, myhubby, myplace, stage, akkordtour, club720, firenze, purplerain, eafamilytree, 720, momlife, nonni, 30anni, thanksgod, somuchlove, pregnantlife, pregnancy, babybump, pregnantandperfect, babyontheway, pregnant, mdv_style, unmaritofavoloso, dasemprepersempre, mumanddaughter, grandparents

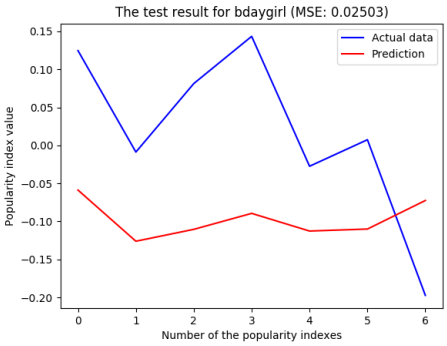
A.2 Results

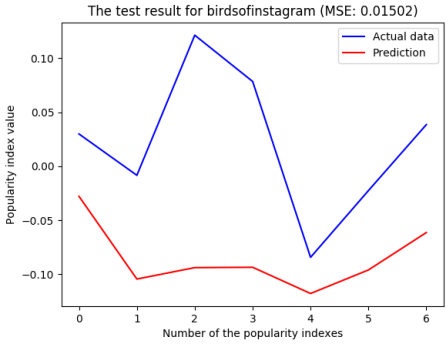
A.2.1 Batch 16, Window 3, Epochs 100

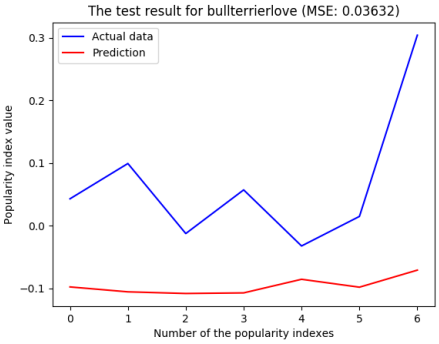


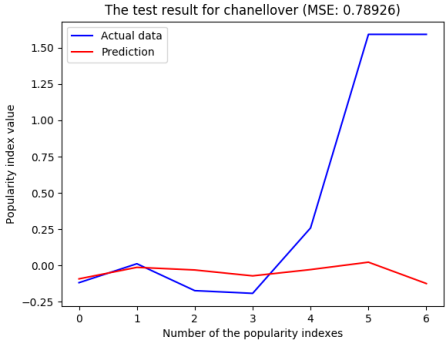


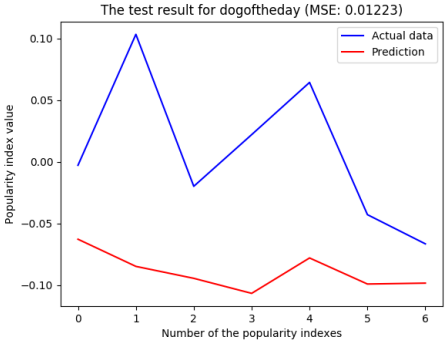




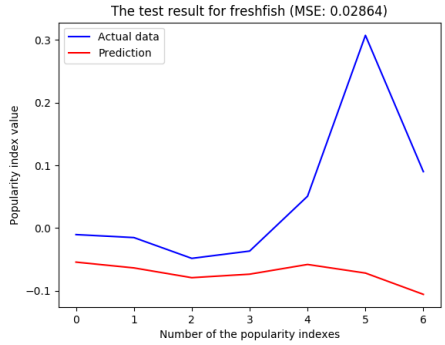


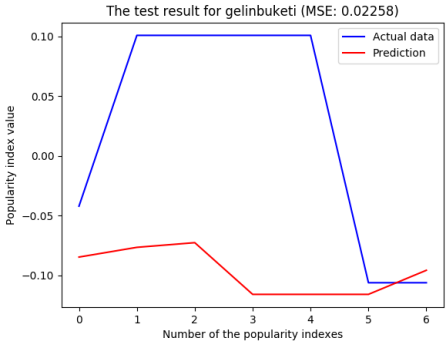


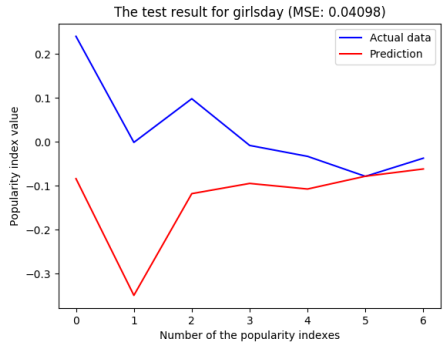


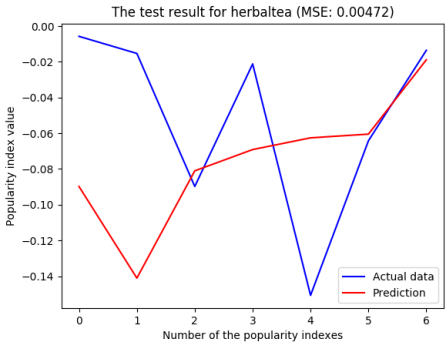


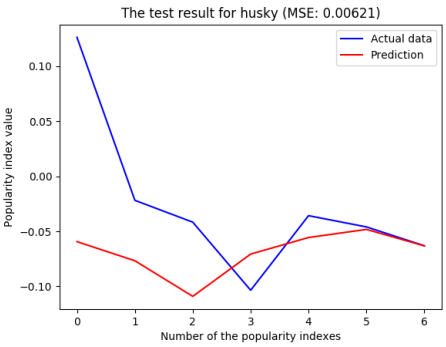
Clusters

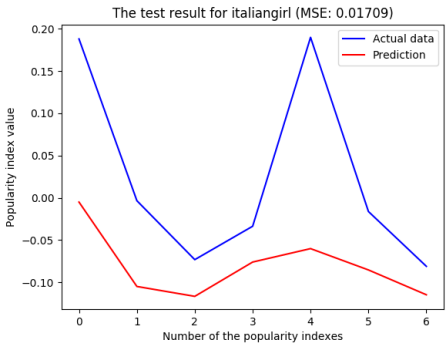

The test result for freshfish (MSE: 0.02864)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

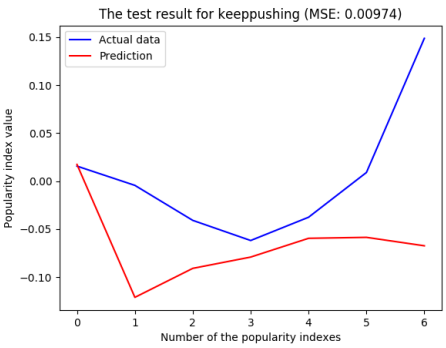

The test result for gelinbuketi (MSE: 0.02258)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

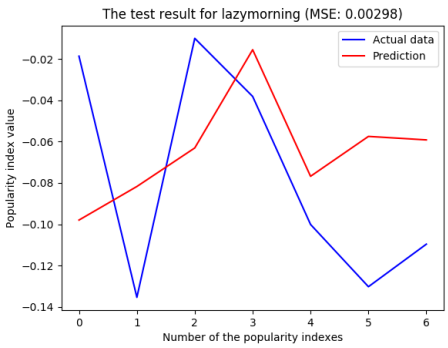

The test result for girlsday (MSE: 0.04098)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

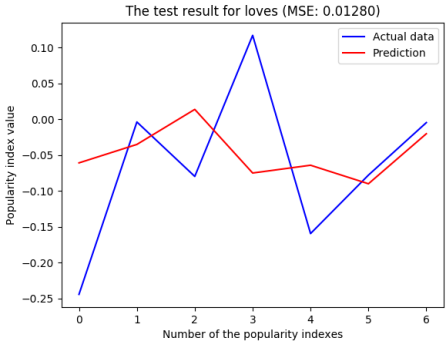

The test result for herbaltea (MSE: 0.00472)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

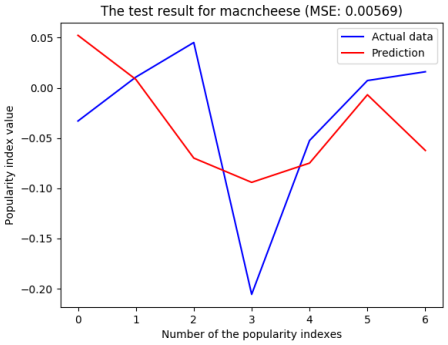

The test result for husky (MSE: 0.00621)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

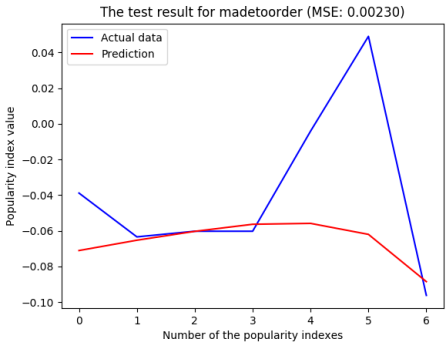

The test result for italiangirl (MSE: 0.01709)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

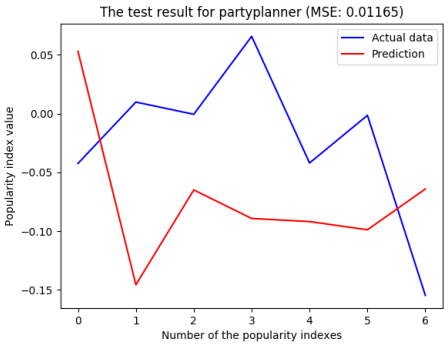

The test result for keeppushing (MSE: 0.00974)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

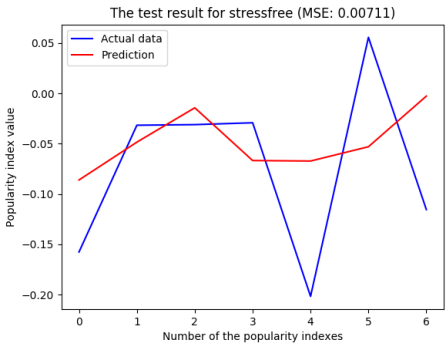

The test result for lazymorning (MSE: 0.00298)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

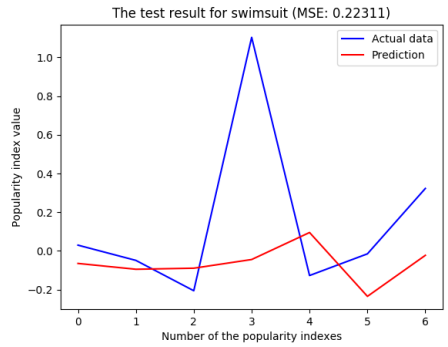

The test result for loves (MSE: 0.01280)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

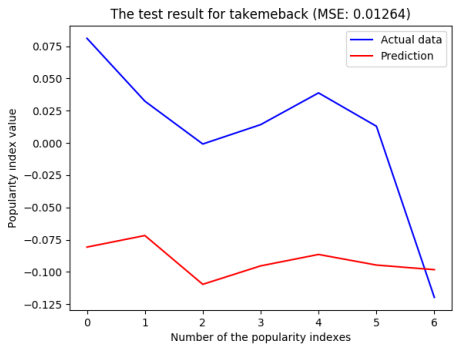

The test result for macncheese (MSE: 0.00569)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

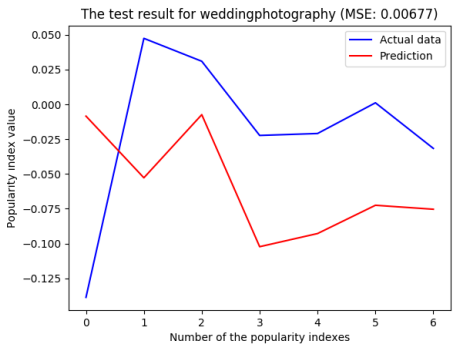

The test result for madetoorder (MSE: 0.00230)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

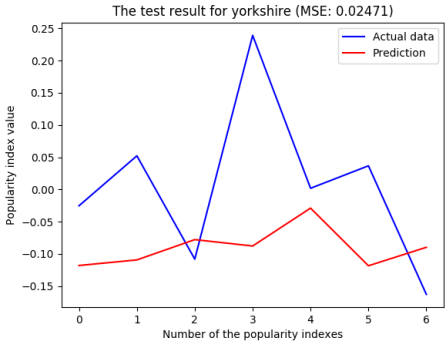

The test result for partyplanner (MSE: 0.01165)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.00711)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

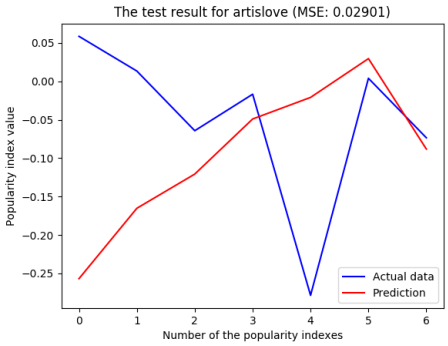

The test result for swimsuit (MSE: 0.22311)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

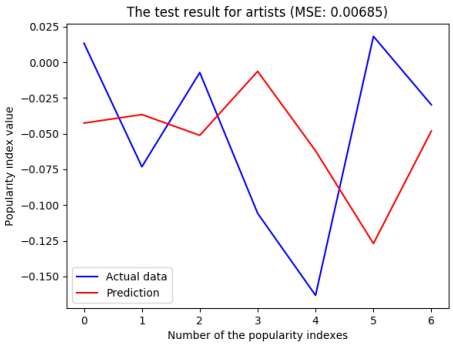

The test result for takemeback (MSE: 0.01264)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

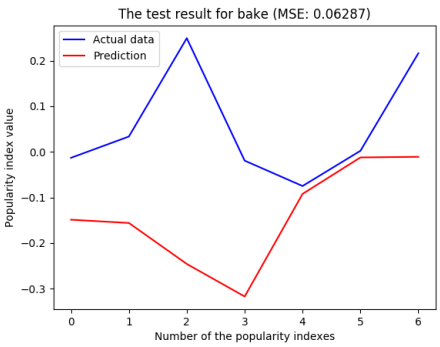

The test result for weddingphotography (MSE: 0.00677)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

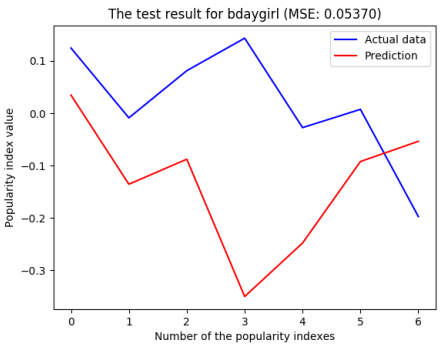


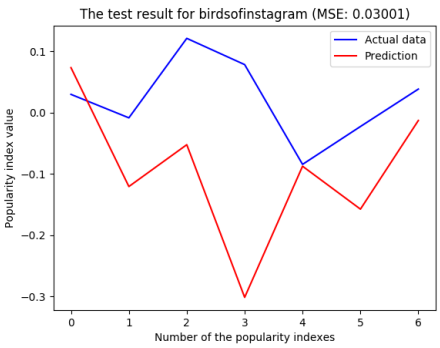
A.2.2 Batch 16, Window 3, Epochs 500

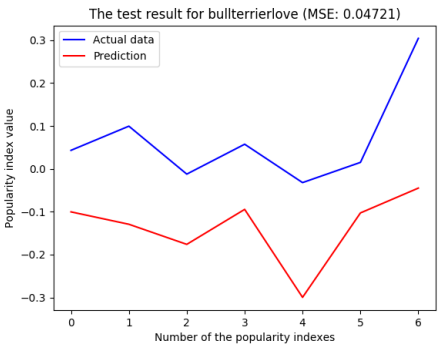




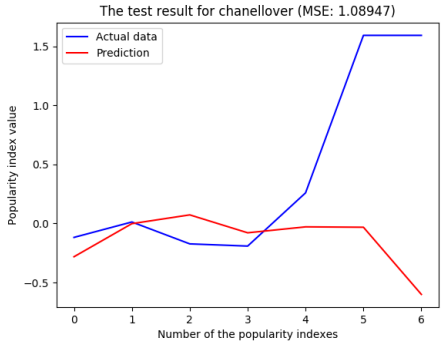


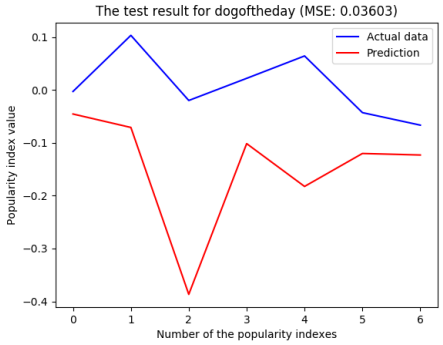


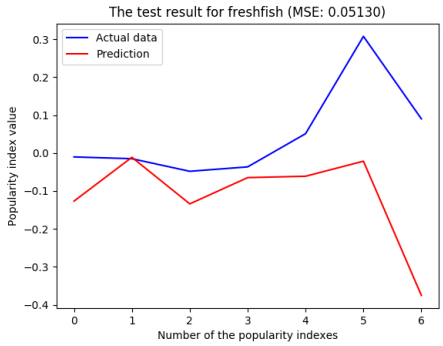


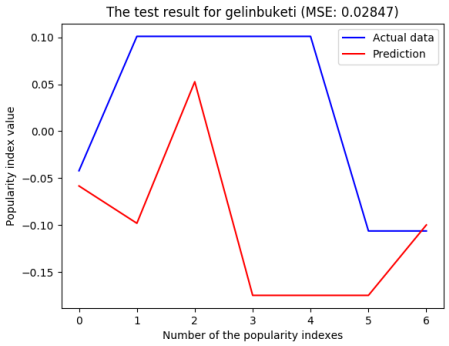


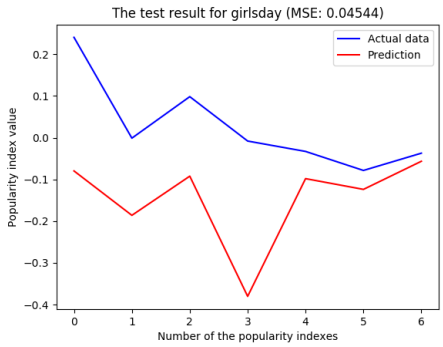
Clusters

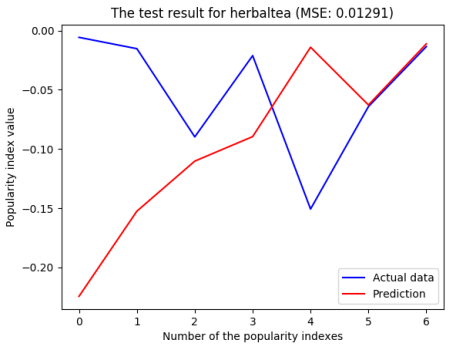

The test result for chanellover (MSE: 1.08947)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

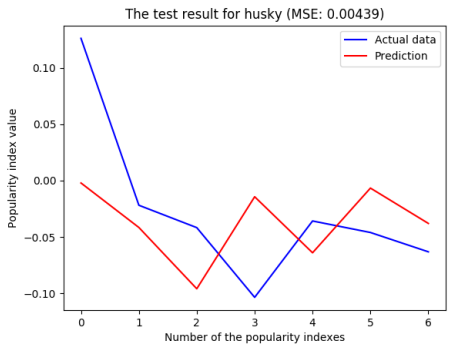

The test result for dogoftheday (MSE: 0.03603)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

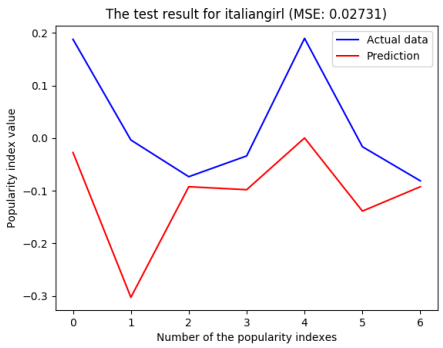

The test result for freshfish (MSE: 0.05130)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

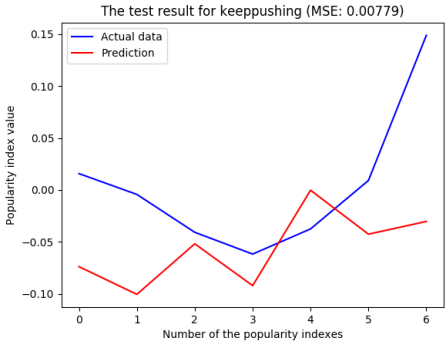

The test result for gelinbuketi (MSE: 0.02847)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

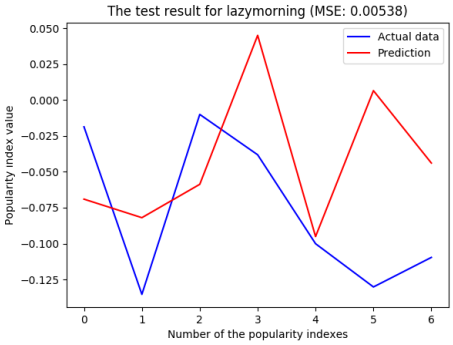

The test result for girlsday (MSE: 0.04544)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

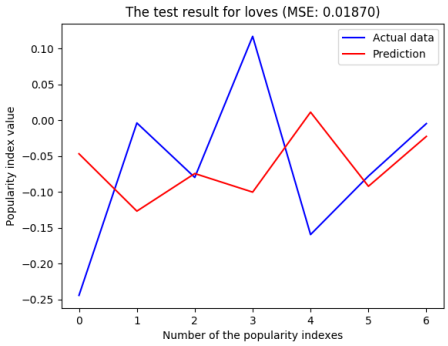

The test result for herbaltea (MSE: 0.01291)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

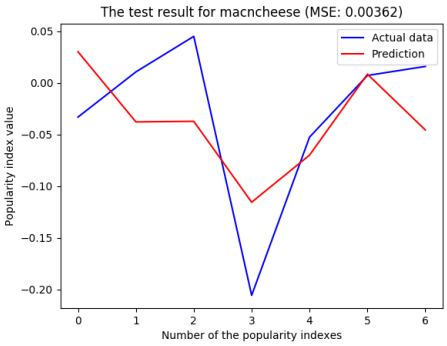

The test result for husky (MSE: 0.00439)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

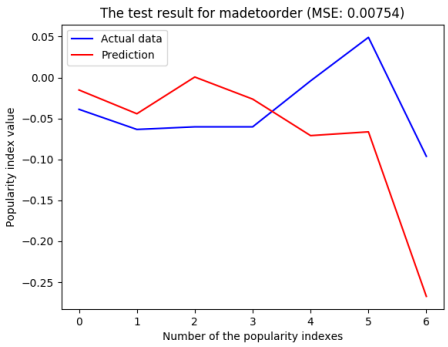

The test result for italiangirl (MSE: 0.02731)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

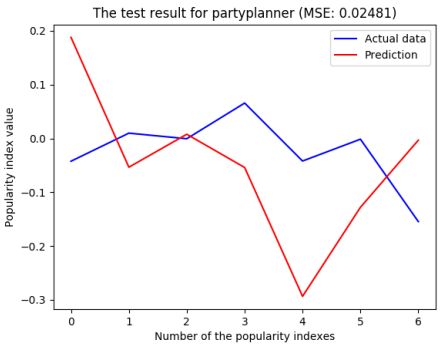

The test result for keeppushing (MSE: 0.00779)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

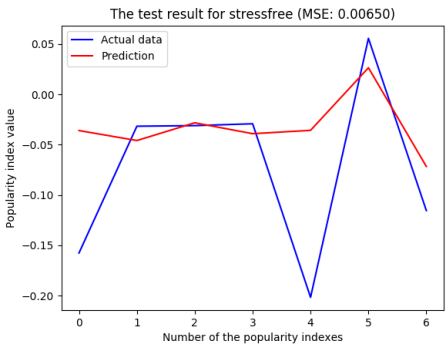

The test result for lazymorning (MSE: 0.00538)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

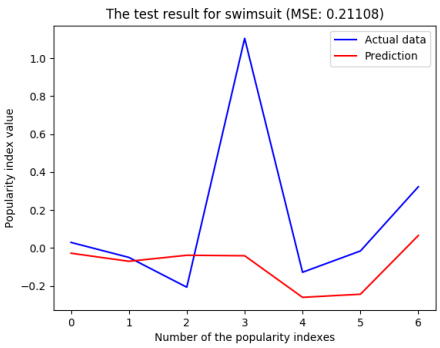

The test result for loves (MSE: 0.01870)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

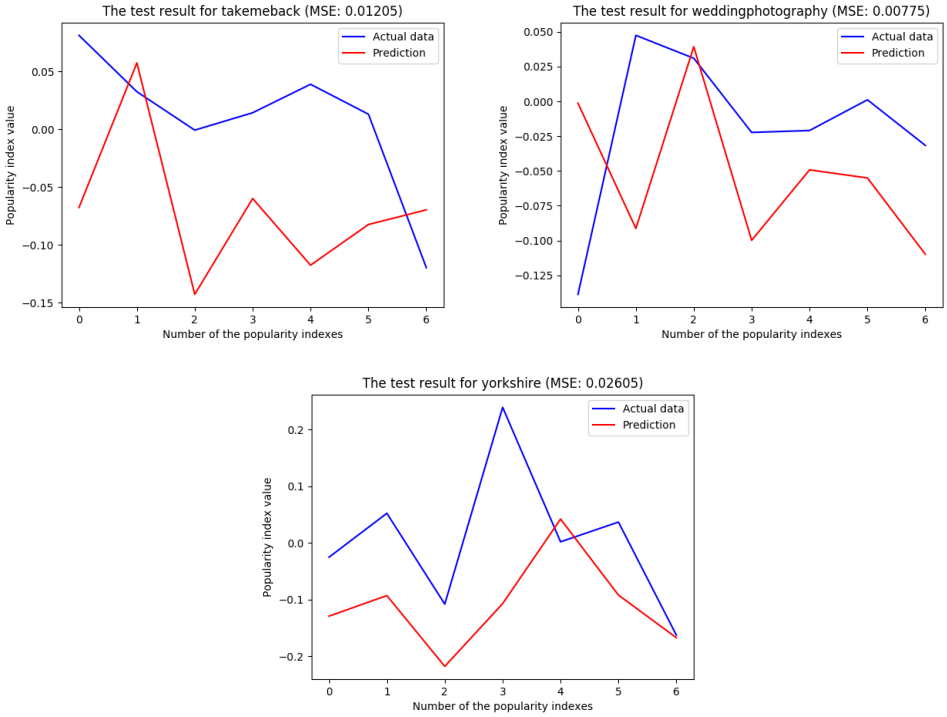

The test result for macncheese (MSE: 0.00362)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00754)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

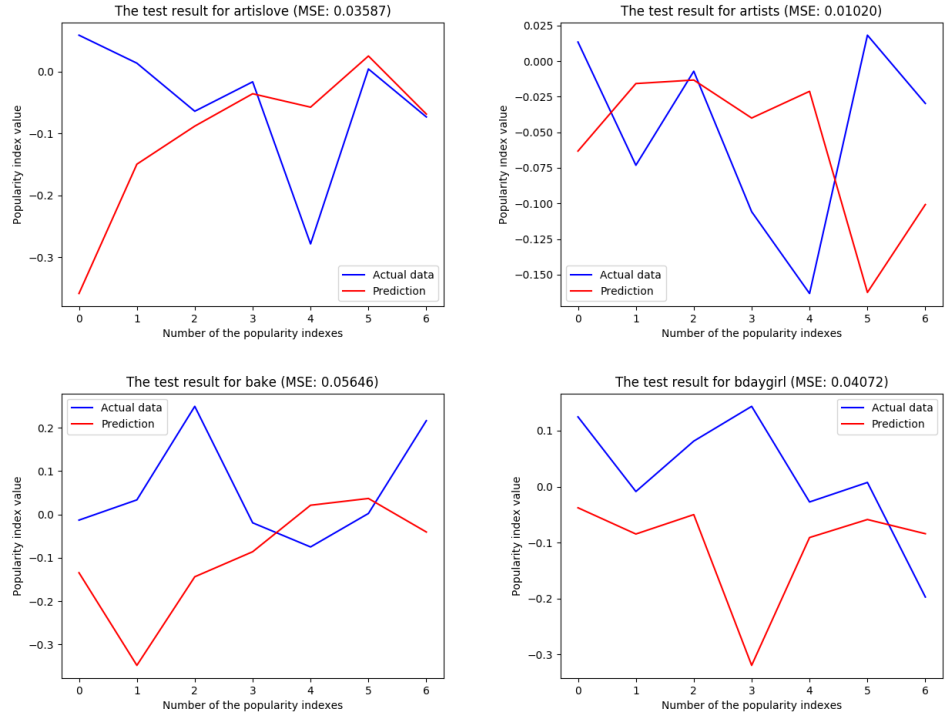

The test result for partyplanner (MSE: 0.02481)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

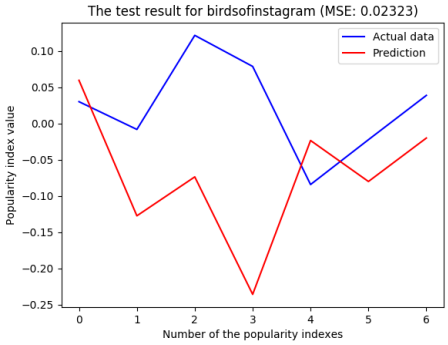

The test result for stressfree (MSE: 0.00650)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

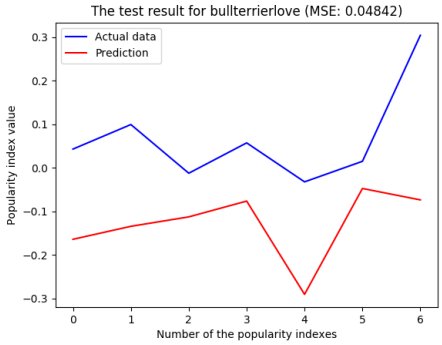

The test result for swimsuit (MSE: 0.21108)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

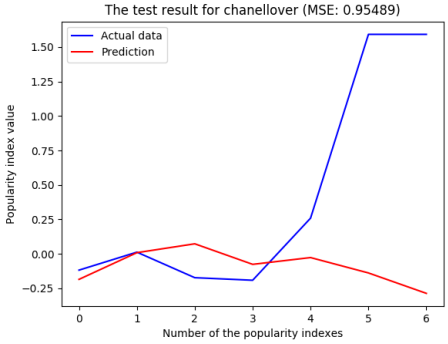


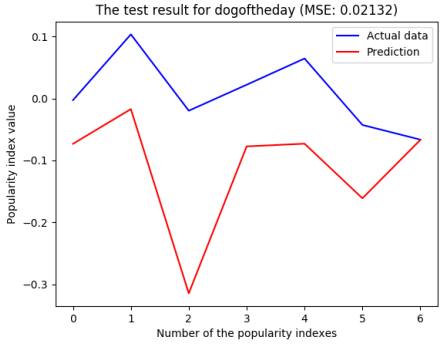
A.2.3 Batch 16, Window 3, Epochs 1000

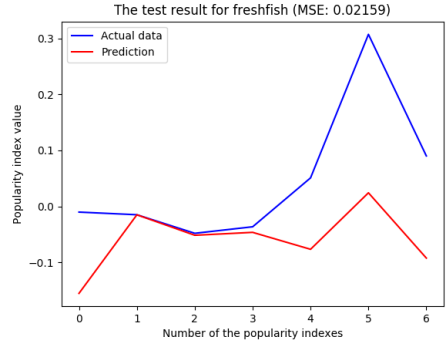


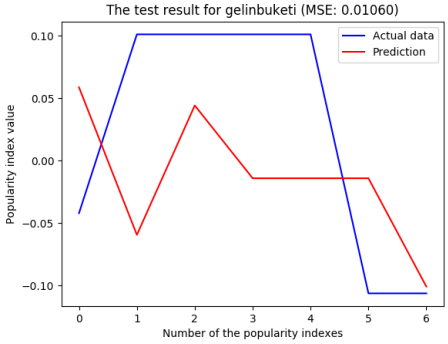

The test result for birdsofinstagram (MSE: 0.02323)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

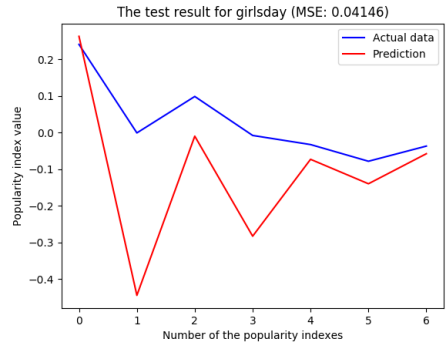

The test result for bullterrierlove (MSE: 0.04842)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

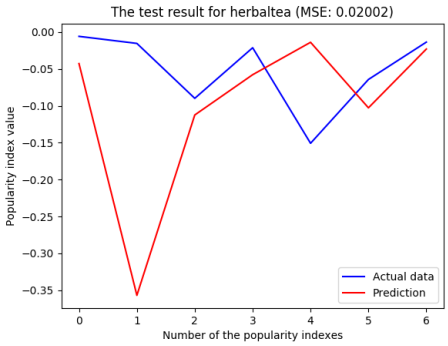

The test result for chanellover (MSE: 0.95489)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

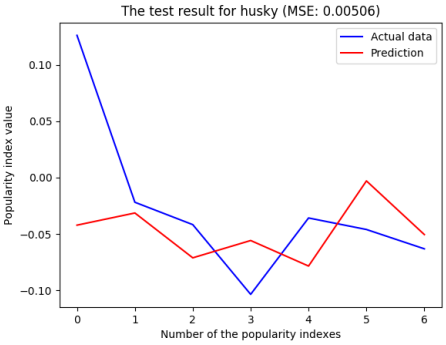

The test result for dogoftheday (MSE: 0.02132)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

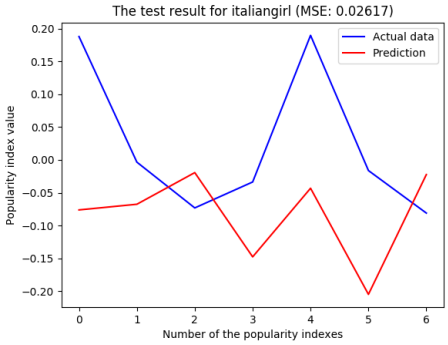

The test result for freshfish (MSE: 0.02159)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

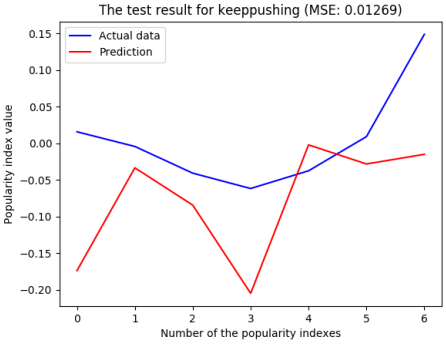

The test result for gelinbuketi (MSE: 0.01060)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

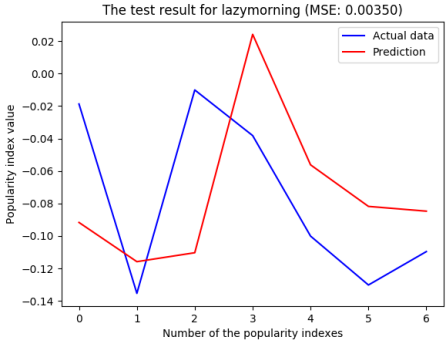

The test result for girlsday (MSE: 0.04146)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

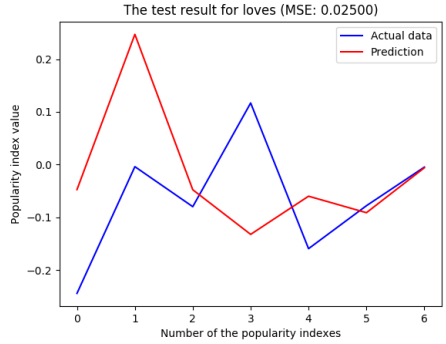

The test result for herbaltea (MSE: 0.02002)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for husky (MSE: 0.00506)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

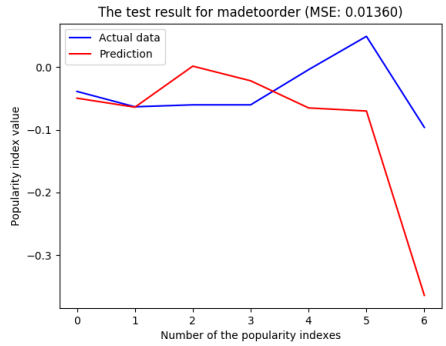

The test result for italiangirl (MSE: 0.02617)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

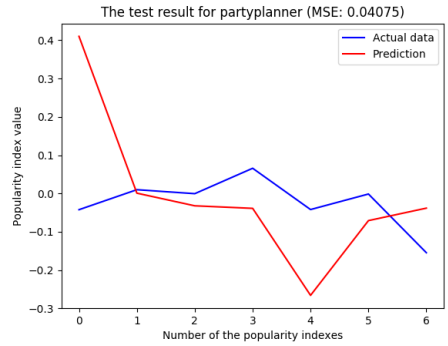

The test result for keeppushing (MSE: 0.01269)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for lazymorning (MSE: 0.00350)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

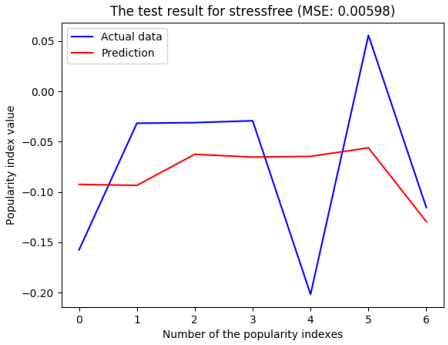

The test result for loves (MSE: 0.02500)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

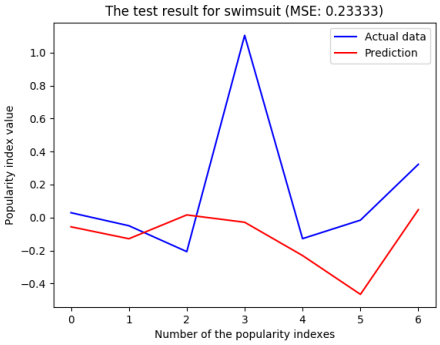

The test result for macncheese (MSE: 0.00455)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

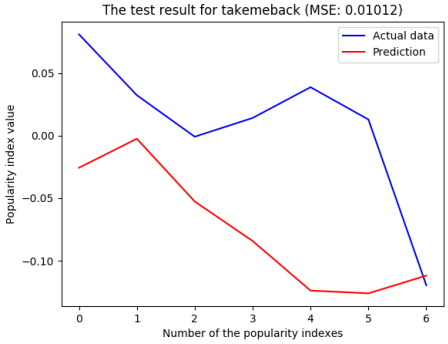

The test result for madetoorder (MSE: 0.01360)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

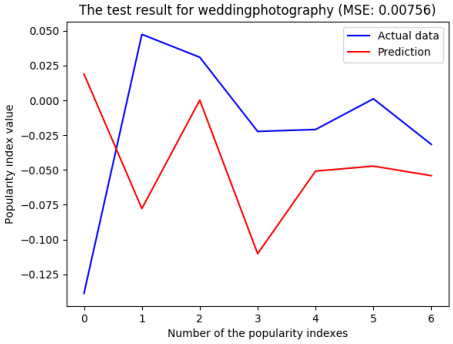

The test result for partyplanner (MSE: 0.04075)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

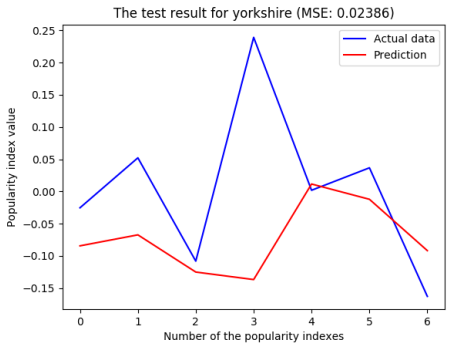
Clusters


The test result for stressfree (MSE: 0.00598)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

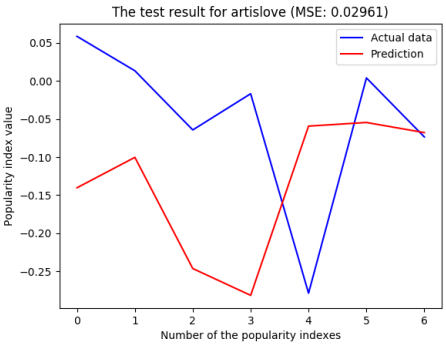

The test result for swimsuit (MSE: 0.23333)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

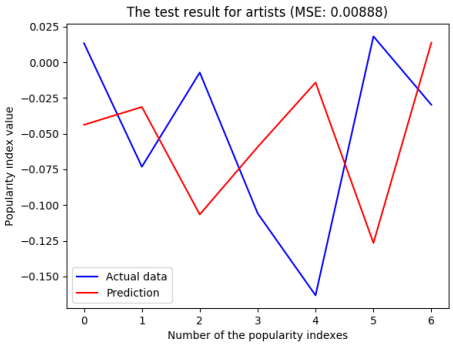

The test result for takemeback (MSE: 0.01012)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

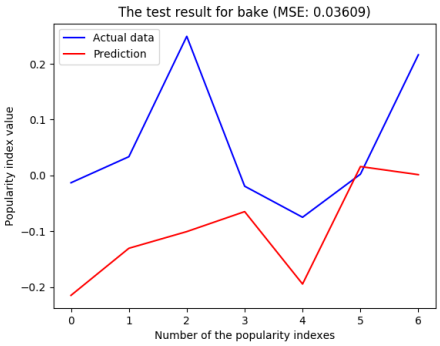

The test result for weddingphotography (MSE: 0.00756)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

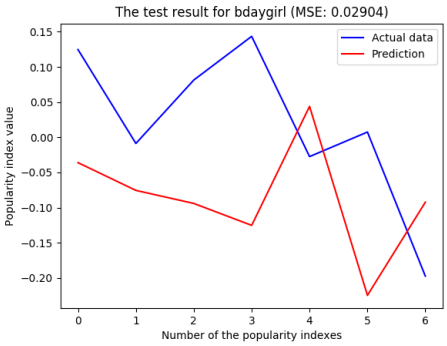

The test result for yorkshire (MSE: 0.02386)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

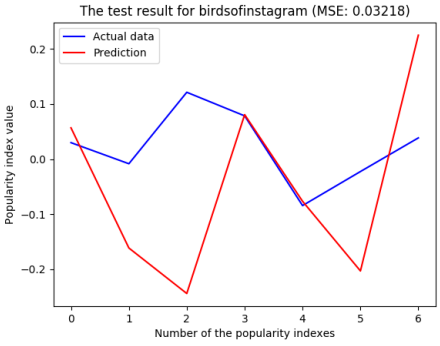
A.2.4 Batch 16, Window 4, Epochs 100

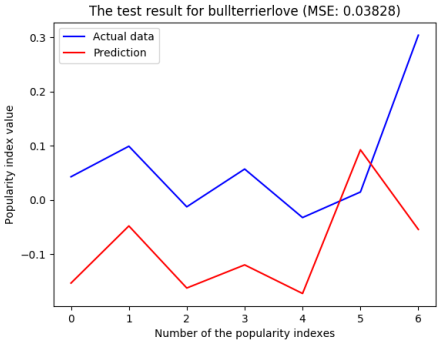


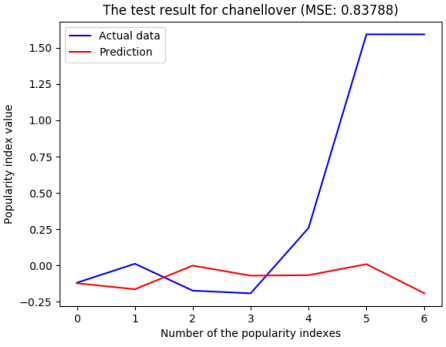


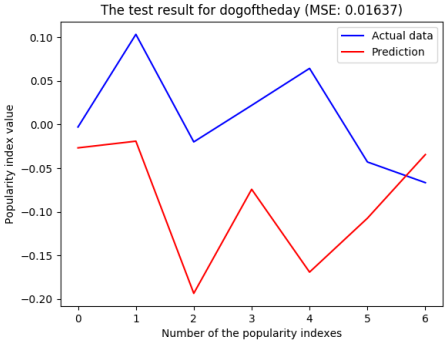


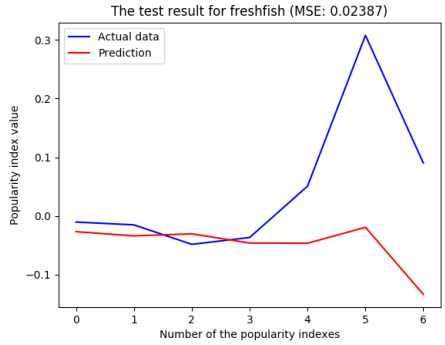


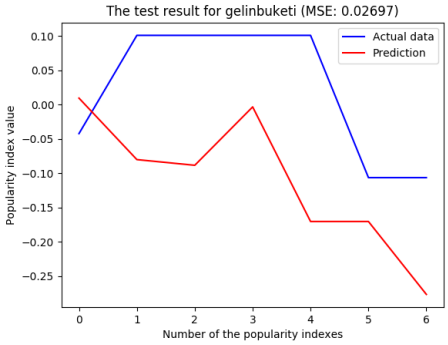


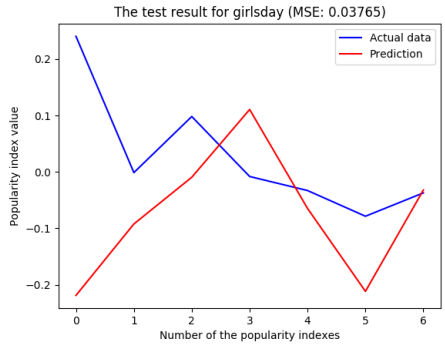


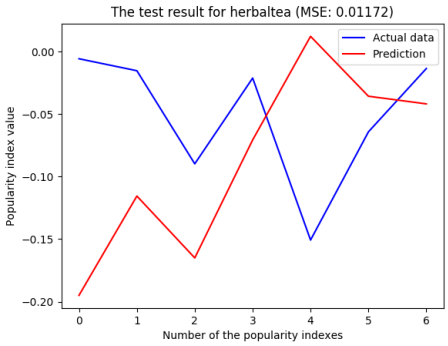


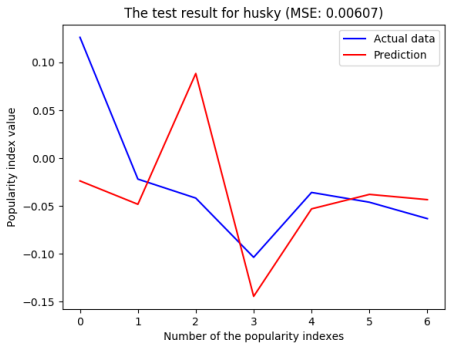


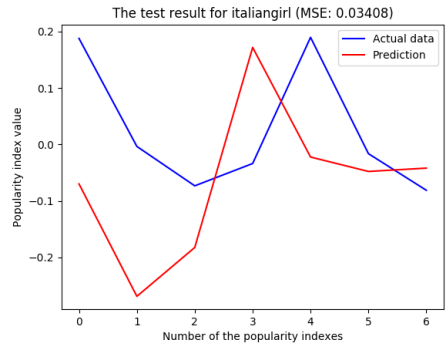

The test result for freshfish (MSE: 0.02387)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

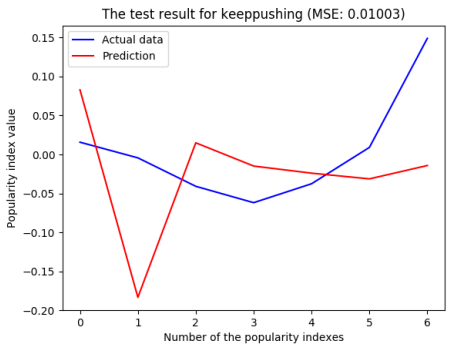

The test result for gelinbuketi (MSE: 0.02697)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

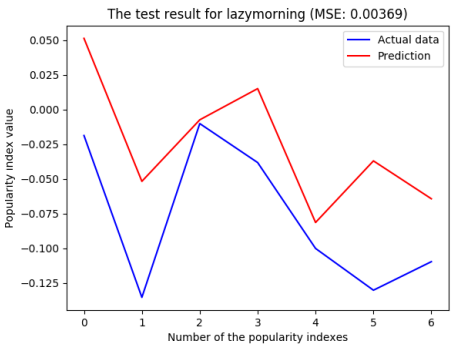

The test result for girlsday (MSE: 0.03765)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for herbaltea (MSE: 0.01172)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

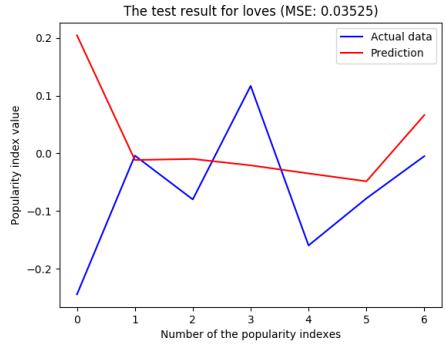

The test result for husky (MSE: 0.00607)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

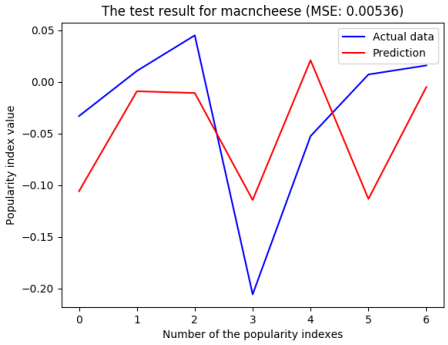

The test result for italiangirl (MSE: 0.03408)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

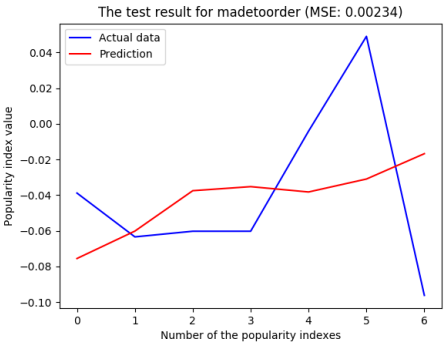

The test result for keeppushing (MSE: 0.01003)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

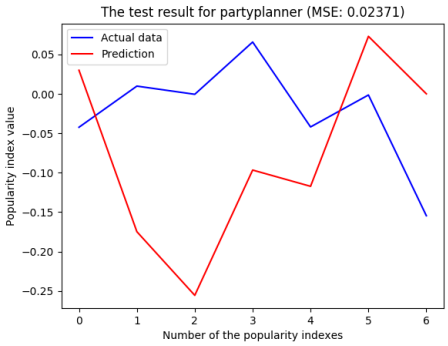

The test result for lazymorning (MSE: 0.00369)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

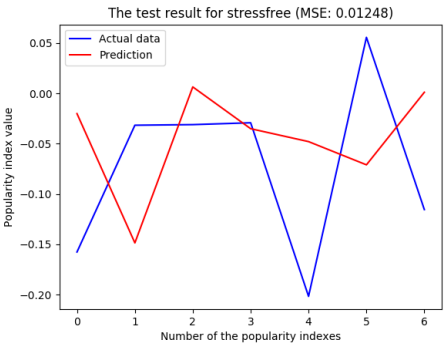
Clusters

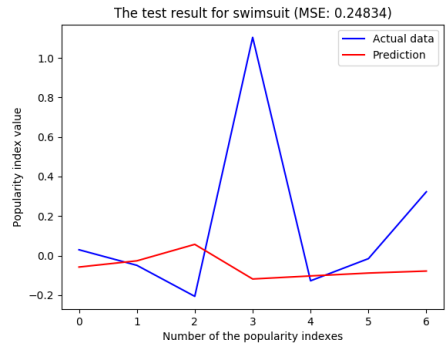

The test result for loves (MSE: 0.03525)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

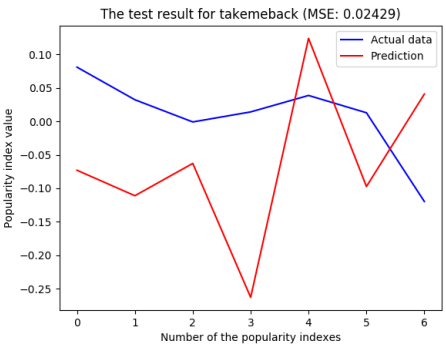

The test result for macncheese (MSE: 0.00536)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

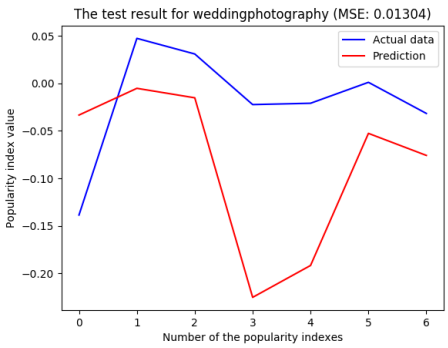

The test result for madetoorder (MSE: 0.00234)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

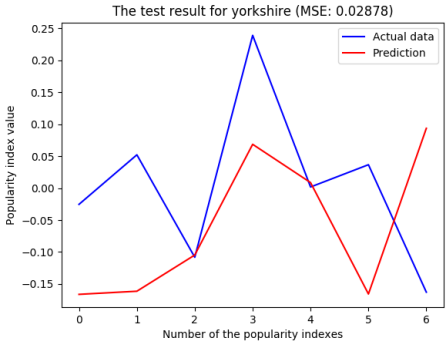

The test result for partyplanner (MSE: 0.02371)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.01248)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

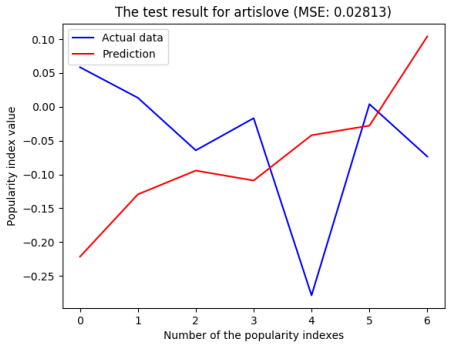

The test result for swimsuit (MSE: 0.24834)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

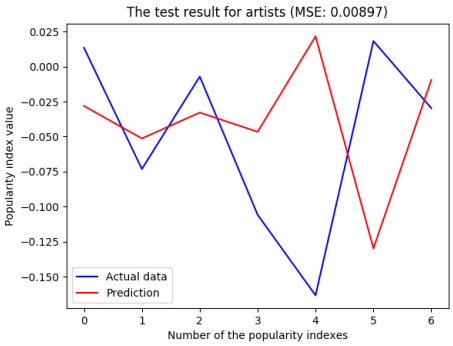

The test result for takemeback (MSE: 0.02429)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

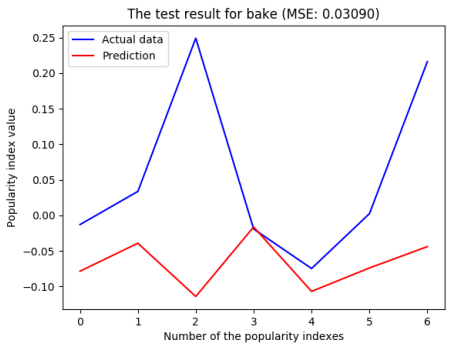

The test result for weddingphotography (MSE: 0.01304)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

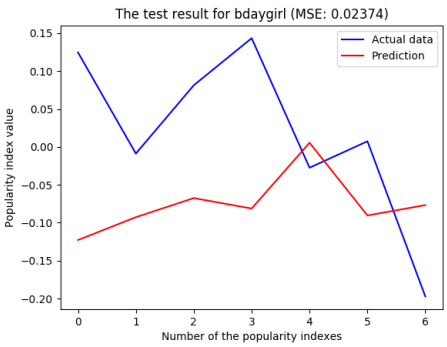


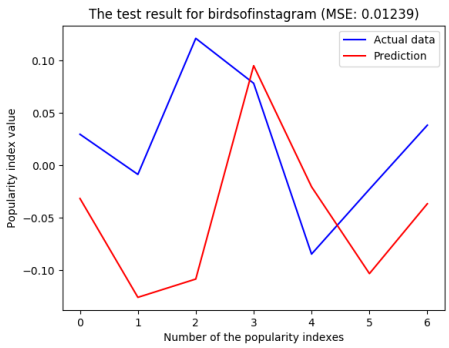
A.2.5 Batch 16, Window 4, Epochs 500

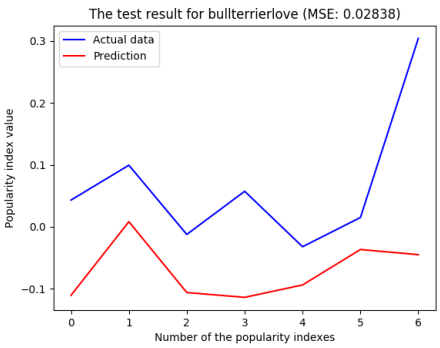




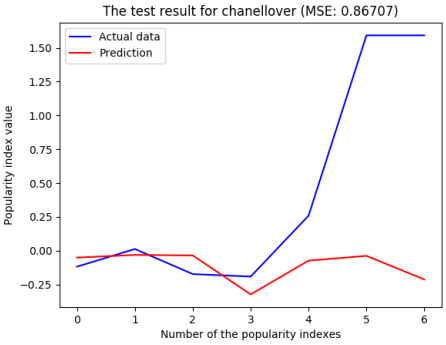


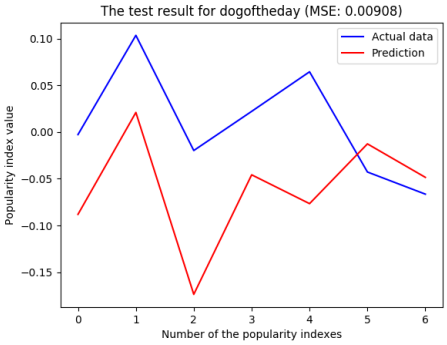


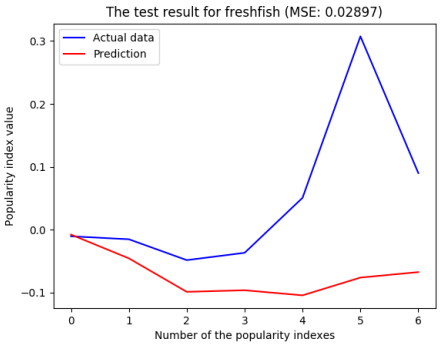


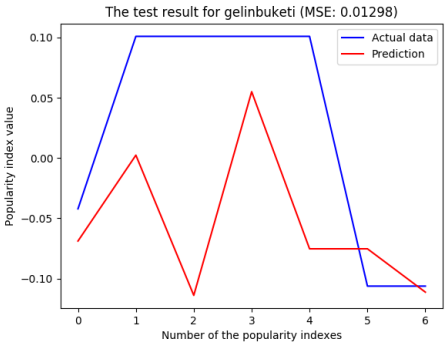


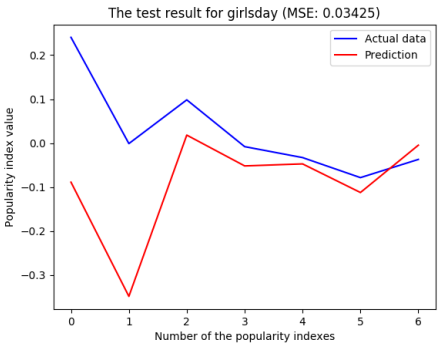
Clusters

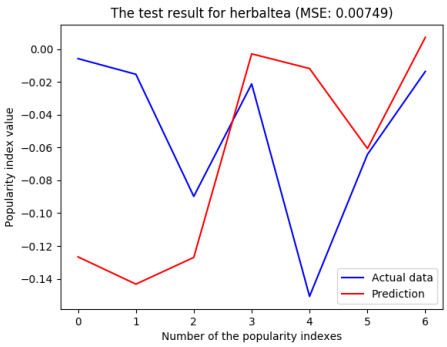

The test result for chanellover (MSE: 0.86707)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

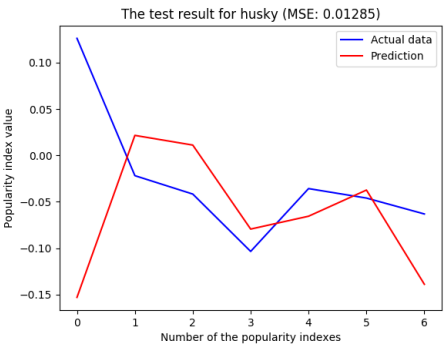

The test result for dogoftheday (MSE: 0.00908)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

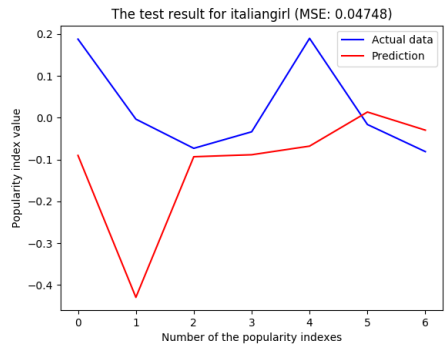

The test result for freshfish (MSE: 0.02897)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

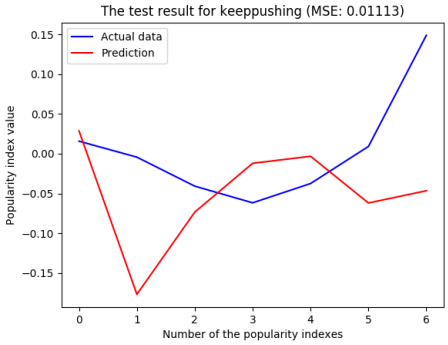

The test result for gelinbuketi (MSE: 0.01298)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

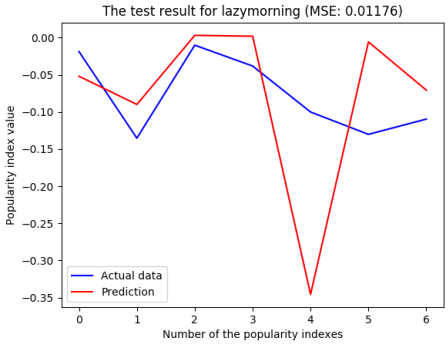

The test result for girlsday (MSE: 0.03425)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

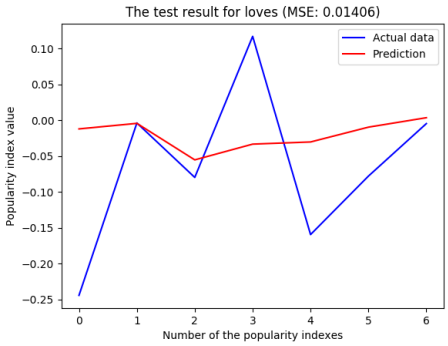

The test result for herbaltea (MSE: 0.00749)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

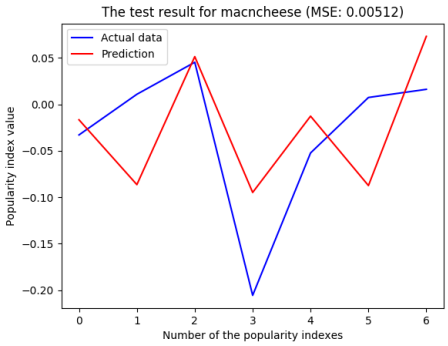

The test result for husky (MSE: 0.01285)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

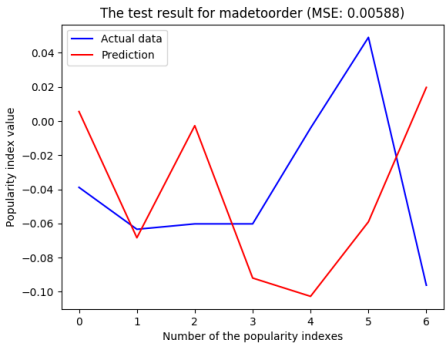

The test result for italiangirl (MSE: 0.04748)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

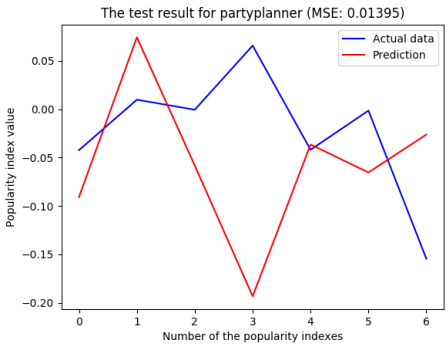

The test result for keeppushing (MSE: 0.01113)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

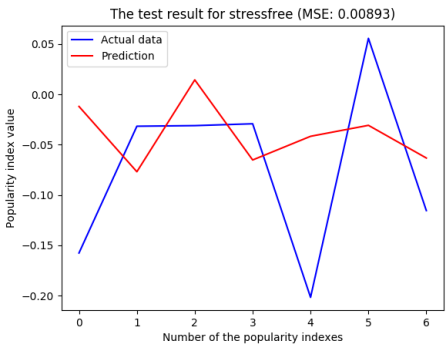

The test result for lazymorning (MSE: 0.01176)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

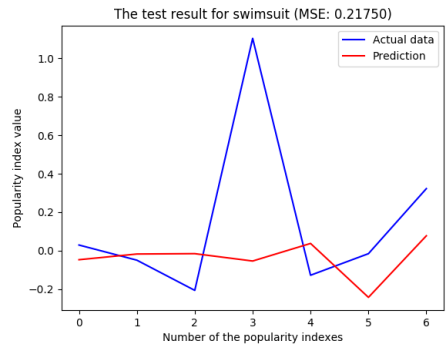

The test result for loves (MSE: 0.01406)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

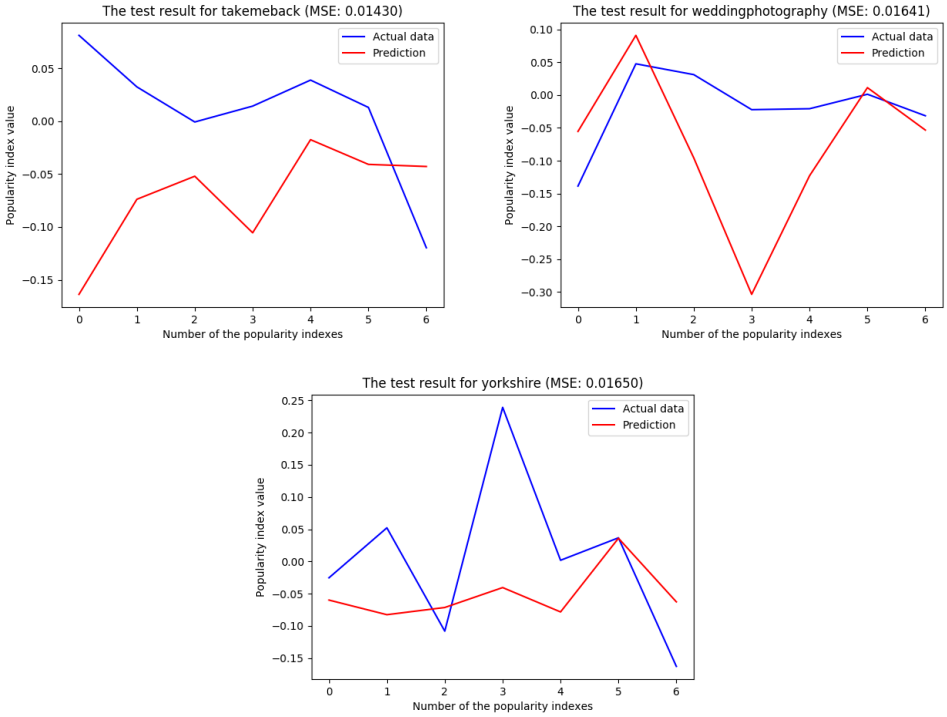

The test result for macncheese (MSE: 0.00512)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00588)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

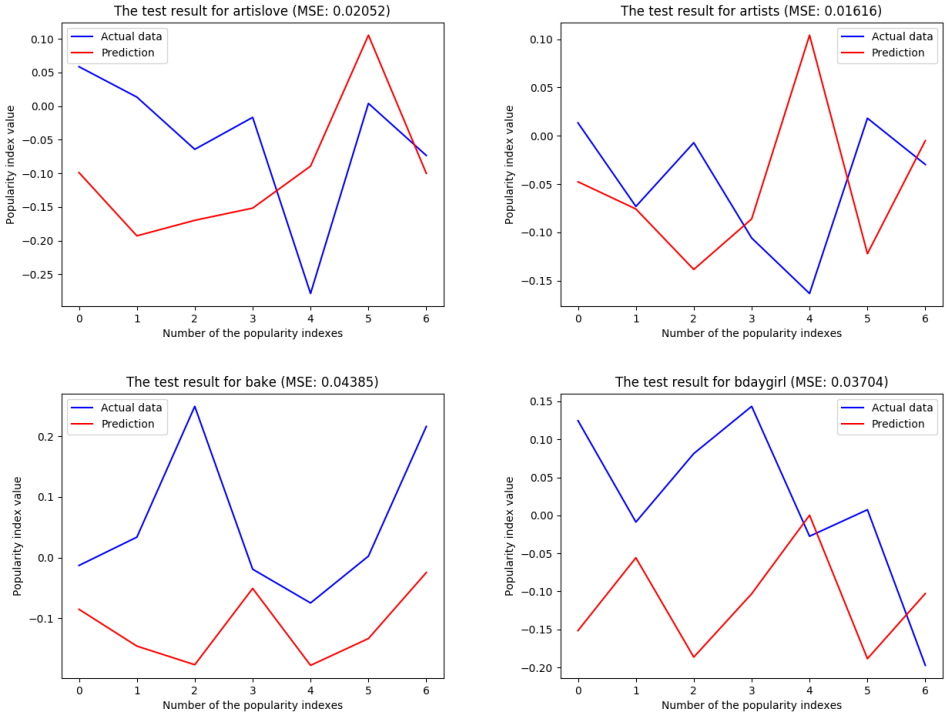

The test result for partyplanner (MSE: 0.01395)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.00893)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

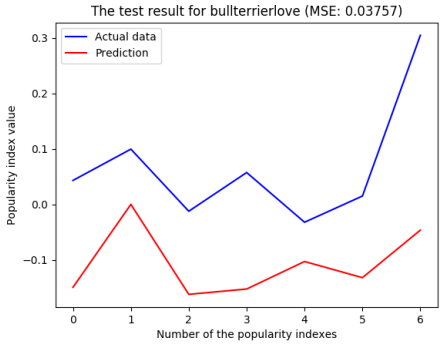

The test result for swimsuit (MSE: 0.21750)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

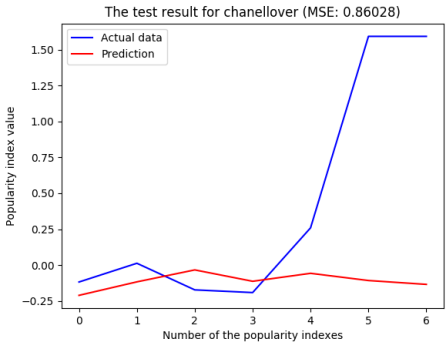


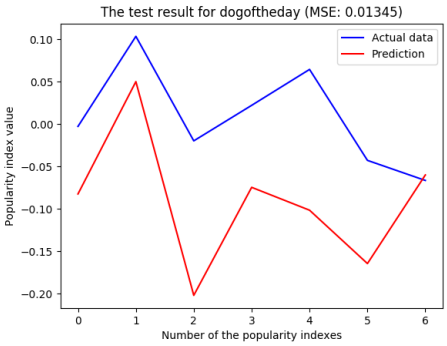
A.2.6 Batch 16, Window 4, Epochs 1000

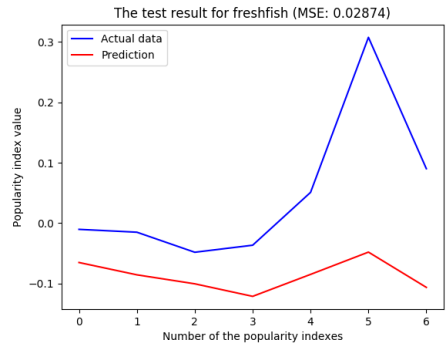


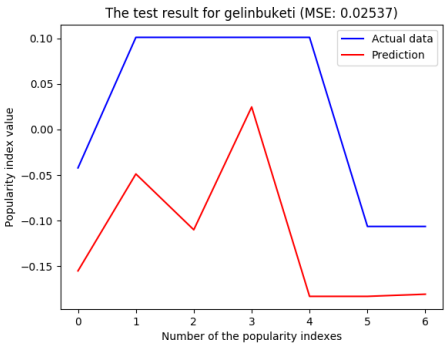

The test result for birdsofinstagram (MSE: 0.01808)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

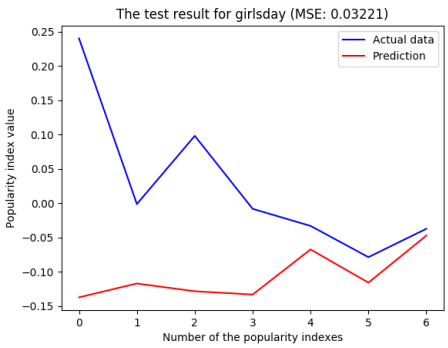

The test result for bullterrierlove (MSE: 0.03757)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

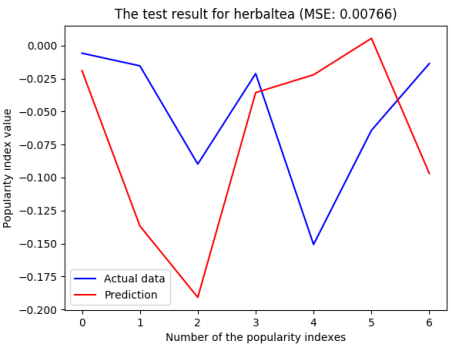

The test result for chanellover (MSE: 0.86028)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

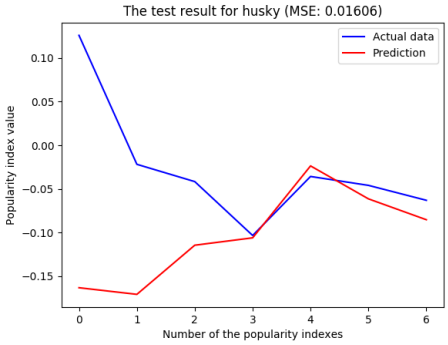

The test result for dogoftheday (MSE: 0.01345)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

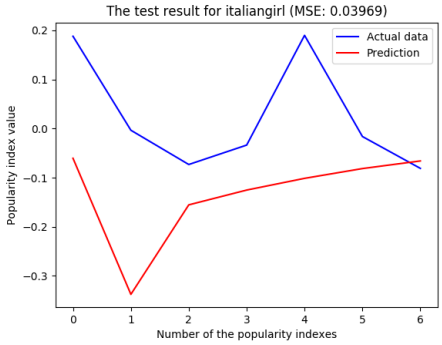

The test result for freshfish (MSE: 0.02874)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

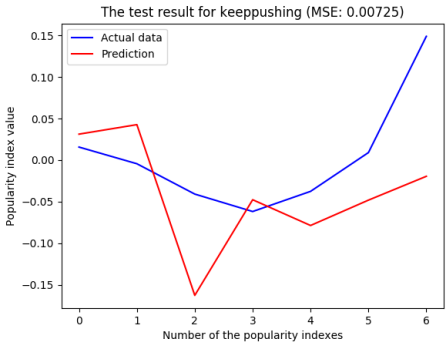

The test result for gelinbuketi (MSE: 0.02537)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

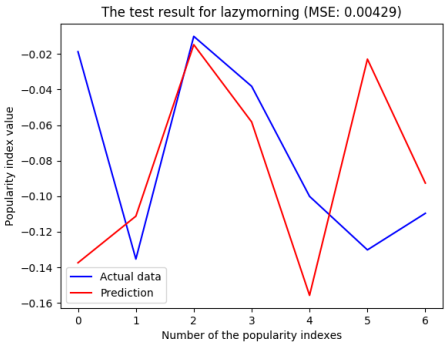

The test result for girlsday (MSE: 0.03221)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for herbaltea (MSE: 0.00766)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

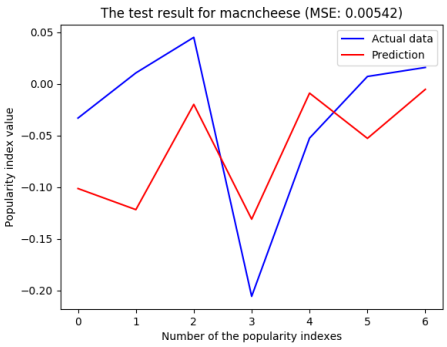

The test result for husky (MSE: 0.01606)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

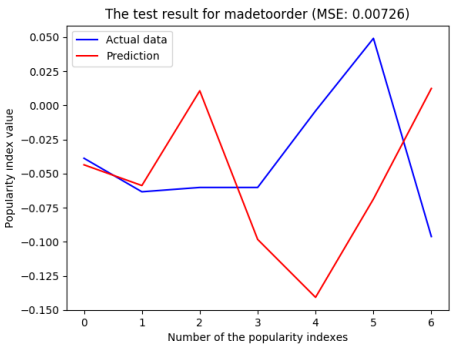

The test result for italiangirl (MSE: 0.03969)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

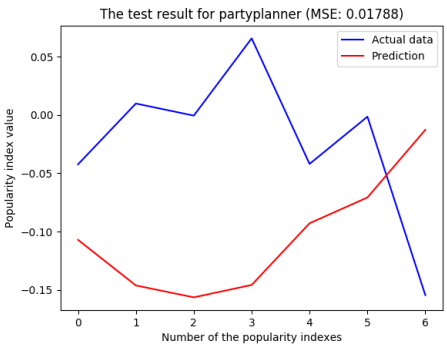

The test result for keeppushing (MSE: 0.00725)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for lazymorning (MSE: 0.00429)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

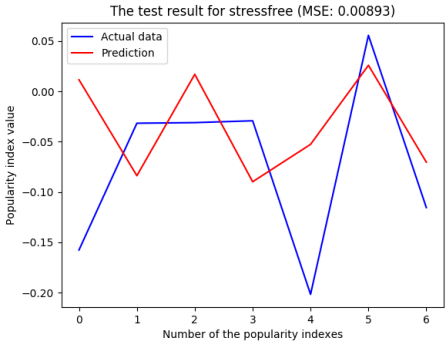

The test result for loves (MSE: 0.02066)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

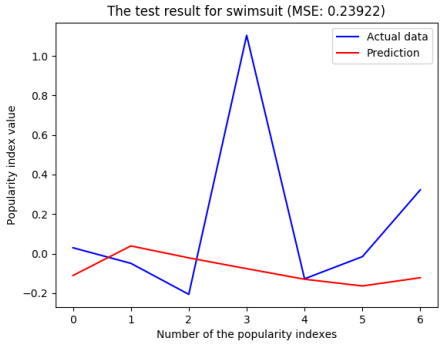

The test result for macncheese (MSE: 0.00542)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

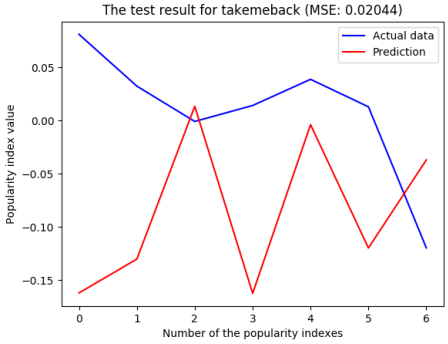

The test result for madetoorder (MSE: 0.00726)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

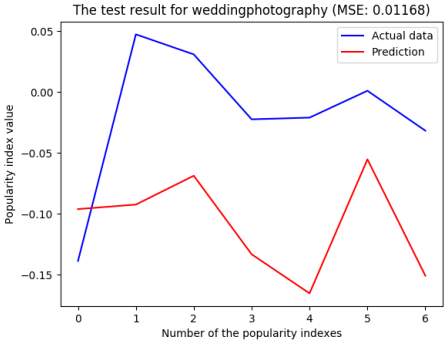

The test result for partyplanner (MSE: 0.01788)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

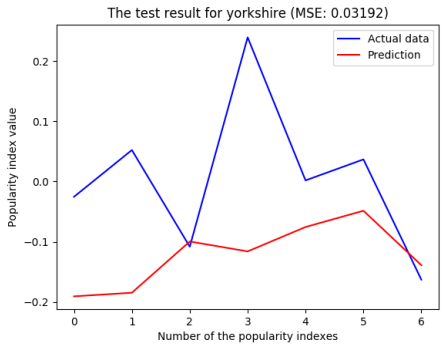
Clusters


The test result for stressfree (MSE: 0.00893)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

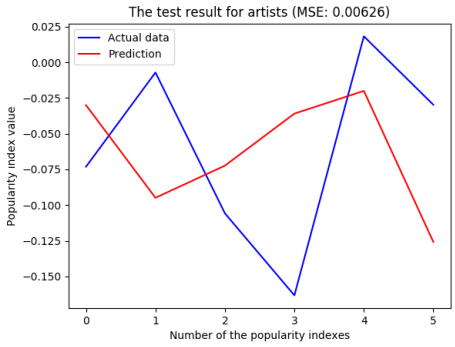

The test result for swimsuit (MSE: 0.23922)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

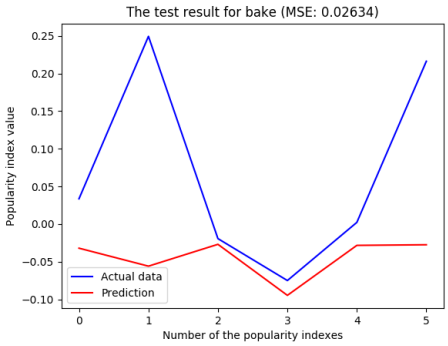

The test result for takemeback (MSE: 0.02044)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

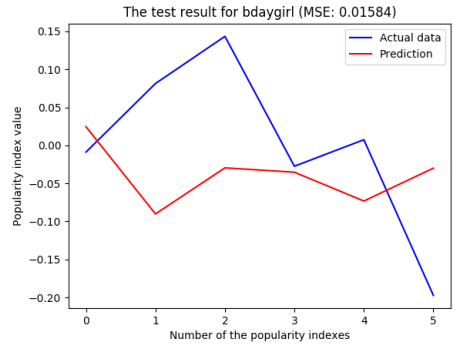

The test result for weddingphotography (MSE: 0.01168)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

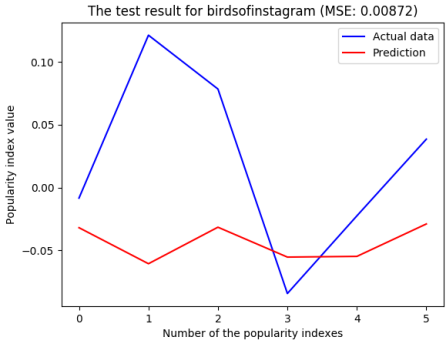

The test result for yorkshire (MSE: 0.03192)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

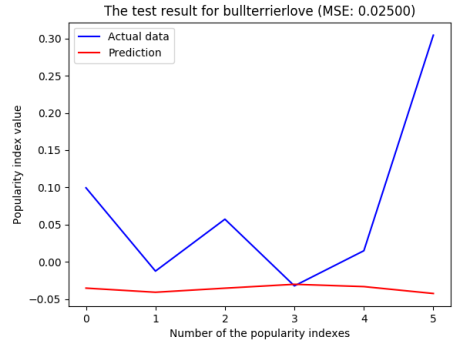
A.2.7 Batch 16, Window 5, Epochs 100

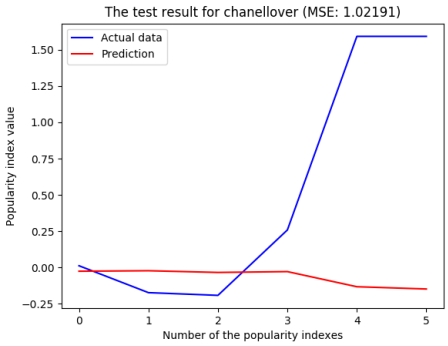


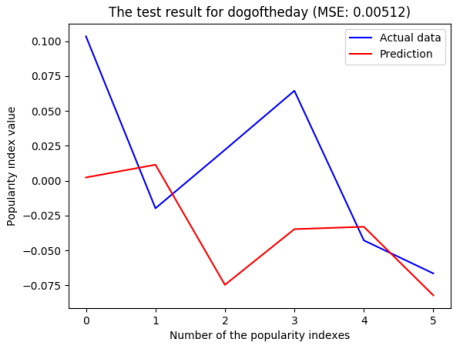


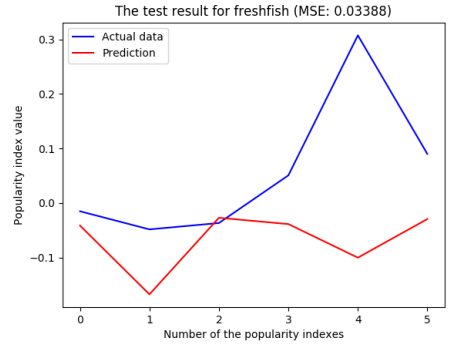


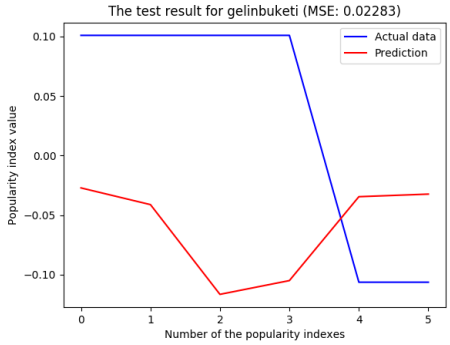


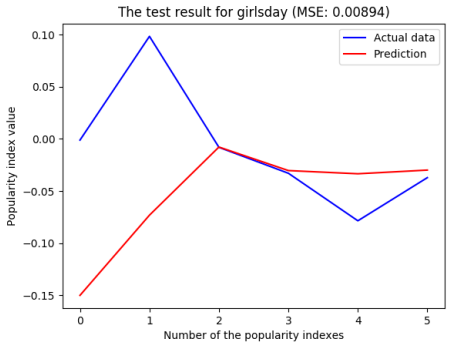


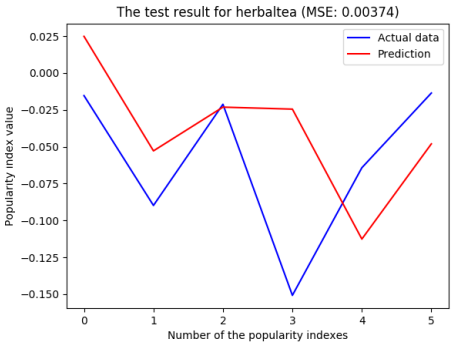


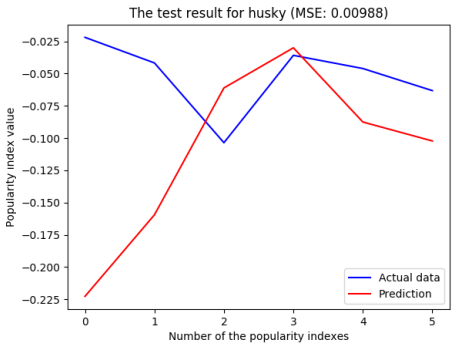


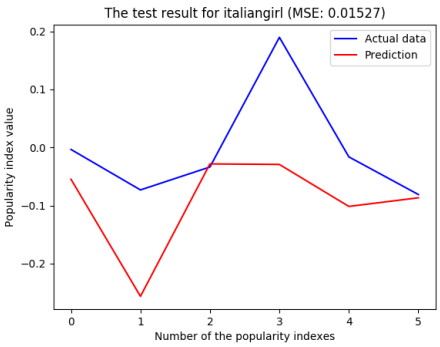

The test result for freshfish (MSE: 0.03388)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

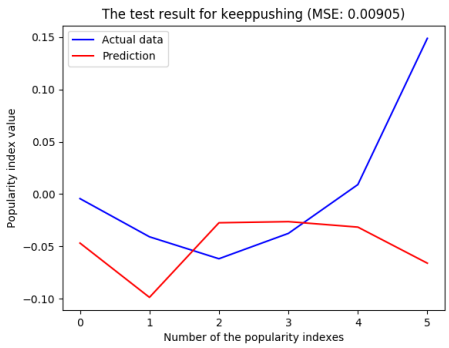

The test result for gelinbuketi (MSE: 0.02283)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

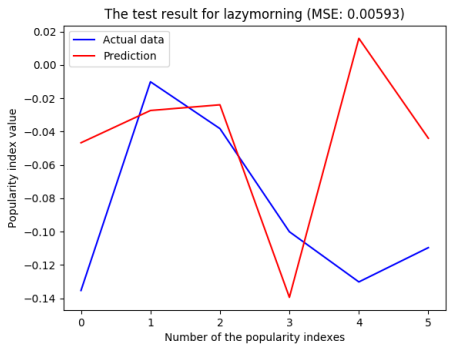

The test result for girlsday (MSE: 0.00894)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for herbaltea (MSE: 0.00374)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

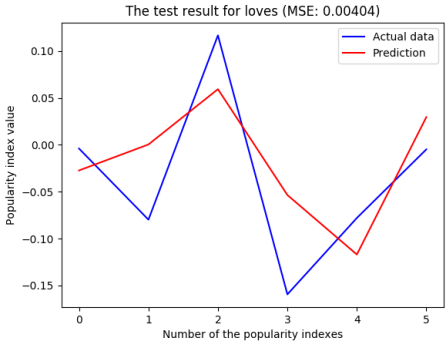

The test result for husky (MSE: 0.00988)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

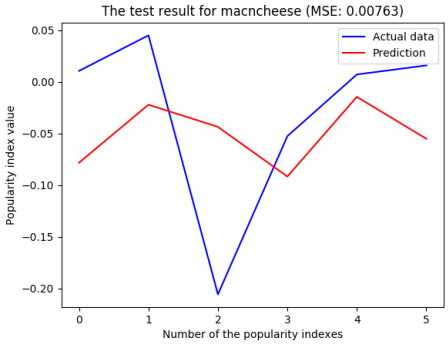

The test result for italiangirl (MSE: 0.01527)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

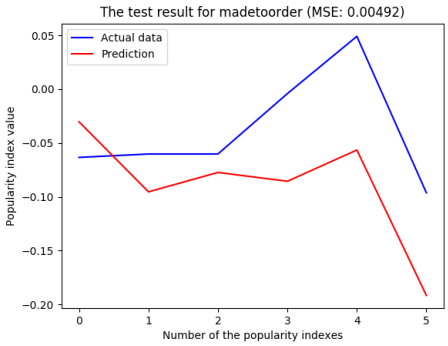

The test result for keeppushing (MSE: 0.00905)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

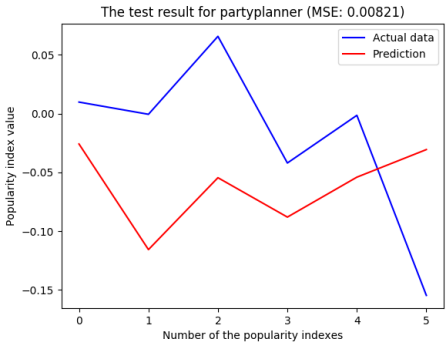

The test result for lazymorning (MSE: 0.00593)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

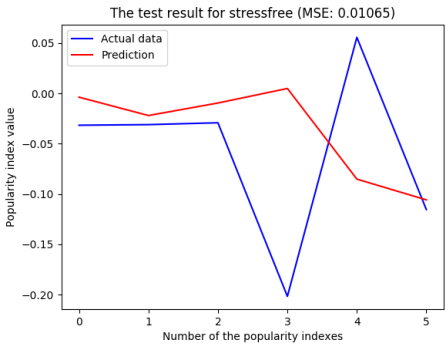
Clusters

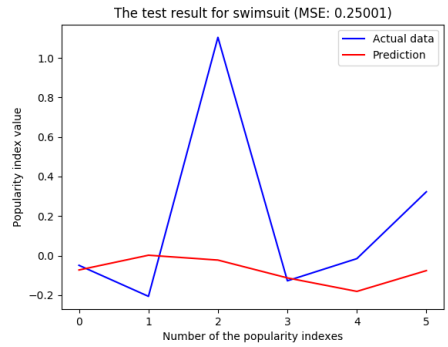

The test result for loves (MSE: 0.00404)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

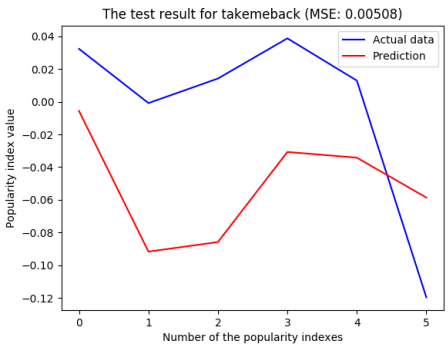

The test result for macncheese (MSE: 0.00763)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

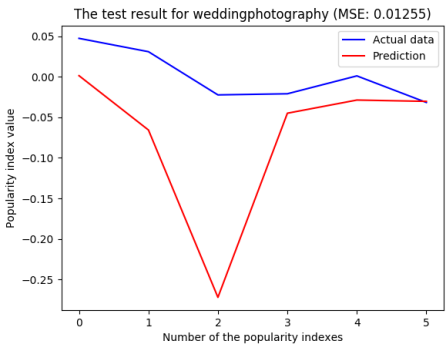

The test result for madetoorder (MSE: 0.00492)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

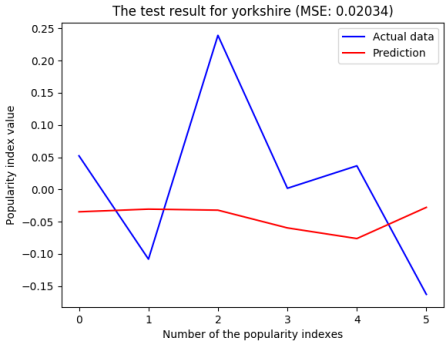

The test result for partyplanner (MSE: 0.00821)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.01065)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

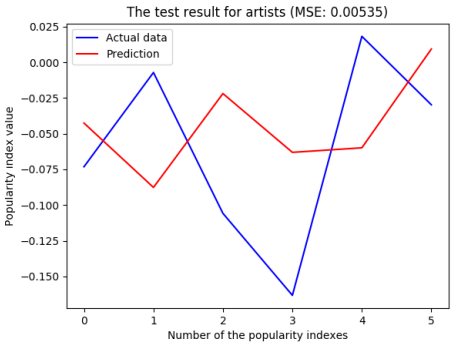

The test result for swimsuit (MSE: 0.25001)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

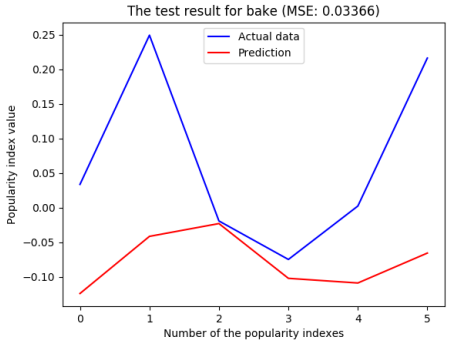

The test result for takemeback (MSE: 0.00508)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

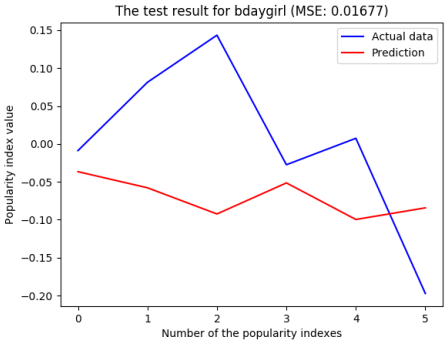

The test result for weddingphotography (MSE: 0.01255)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

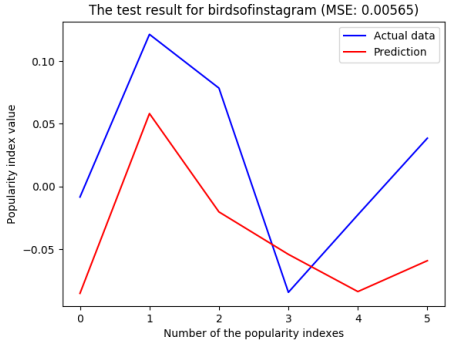


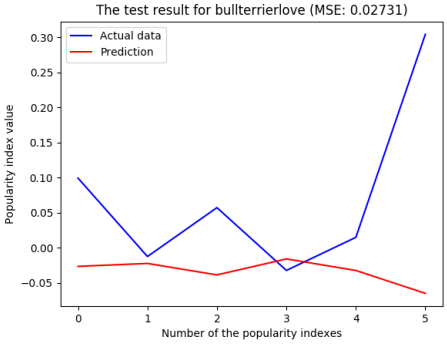
A.2.8 Batch 16, Window 5, Epochs 500



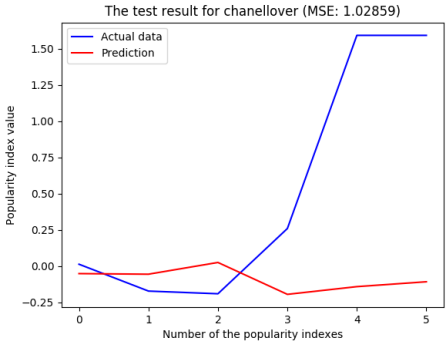


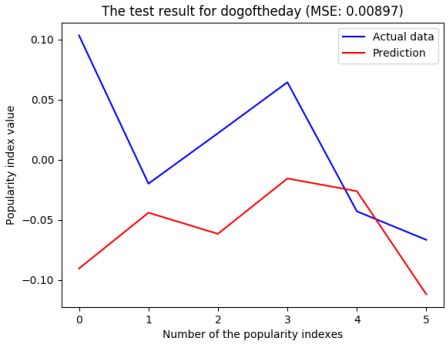


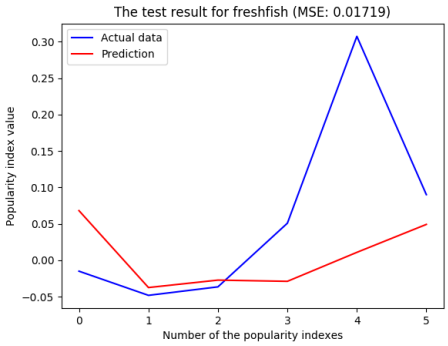


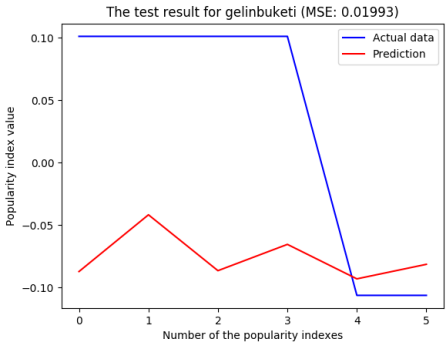


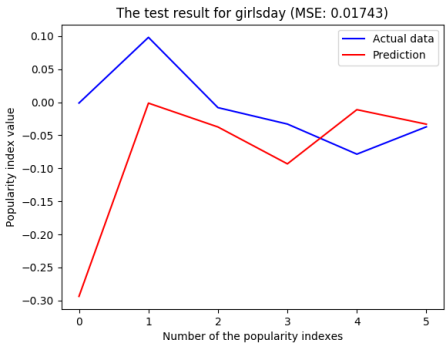
Clusters

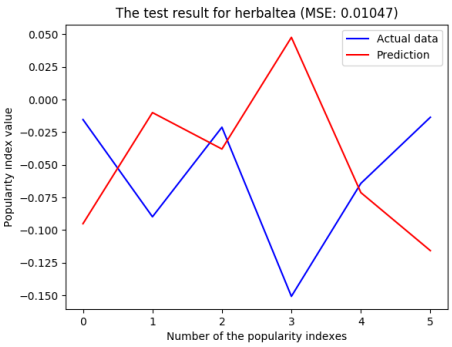

The test result for chanellover (MSE: 1.02859)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

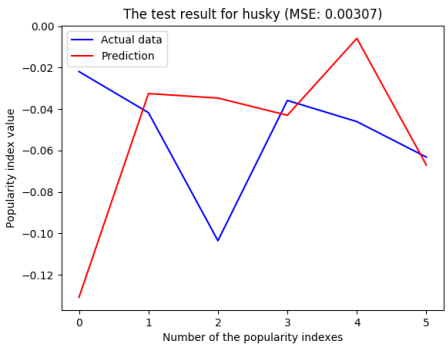

The test result for dogoftheday (MSE: 0.00897)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

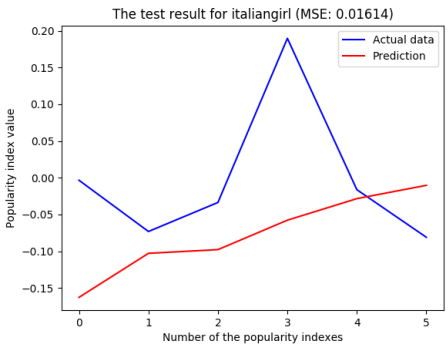

The test result for freshfish (MSE: 0.01719)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

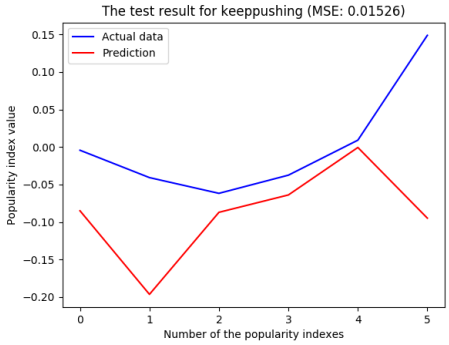

The test result for gelinbuketi (MSE: 0.01993)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

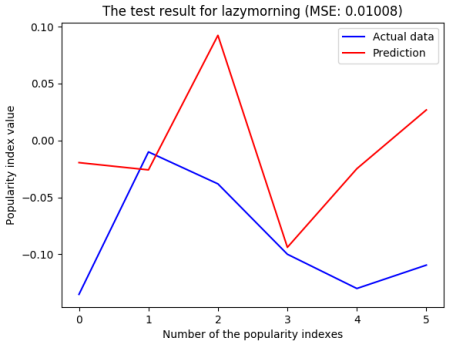

The test result for girlsday (MSE: 0.01743)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

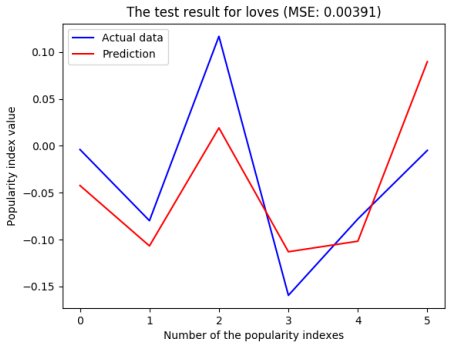

The test result for herbaltea (MSE: 0.01047)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

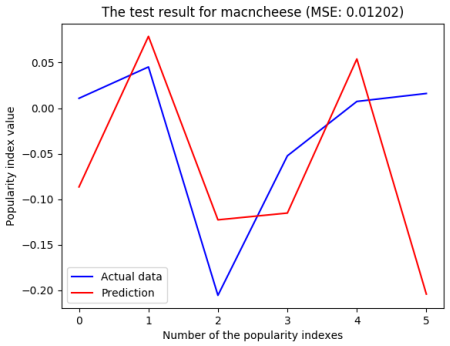

The test result for husky (MSE: 0.00307)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

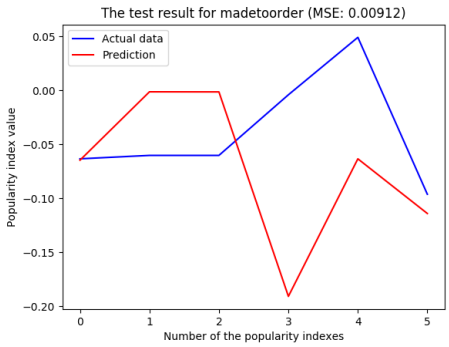

The test result for italiangirl (MSE: 0.01614)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

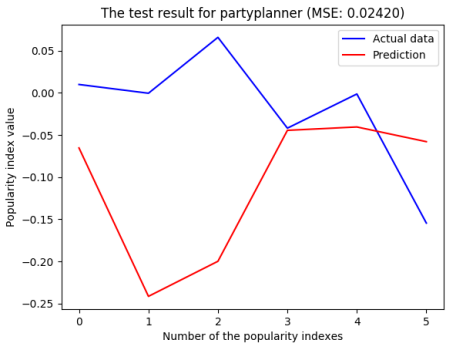

The test result for keeppushing (MSE: 0.01526)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

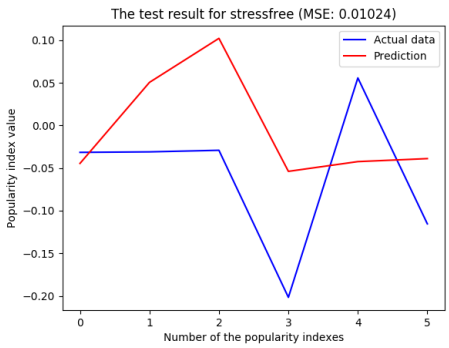

The test result for lazymorning (MSE: 0.01008)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

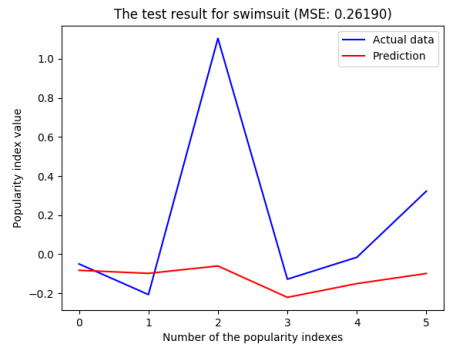

The test result for loves (MSE: 0.00391)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

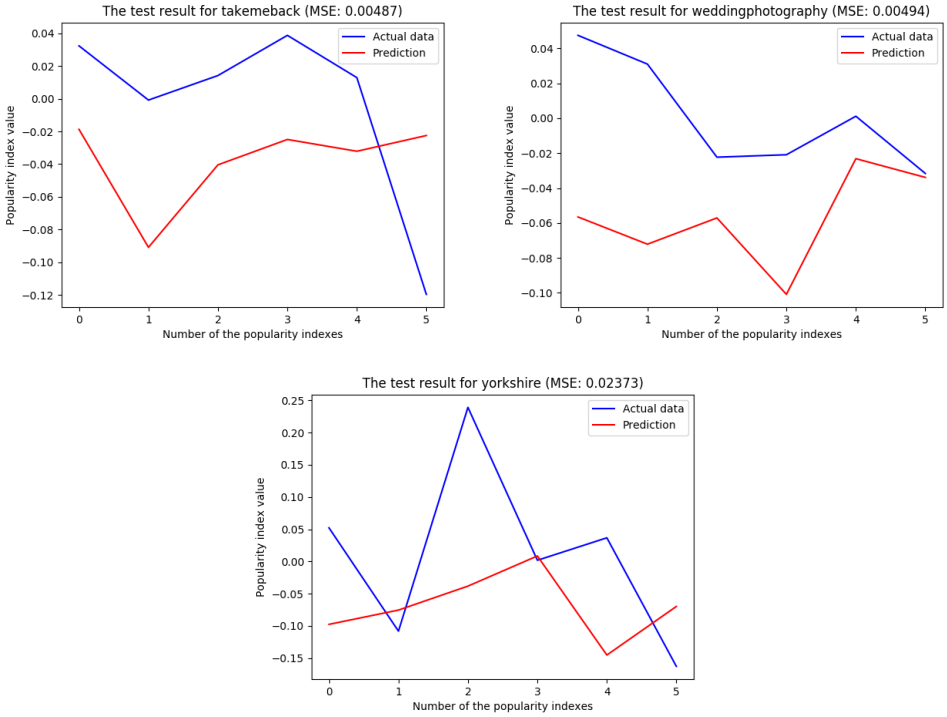

The test result for macncheese (MSE: 0.01202)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00912)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

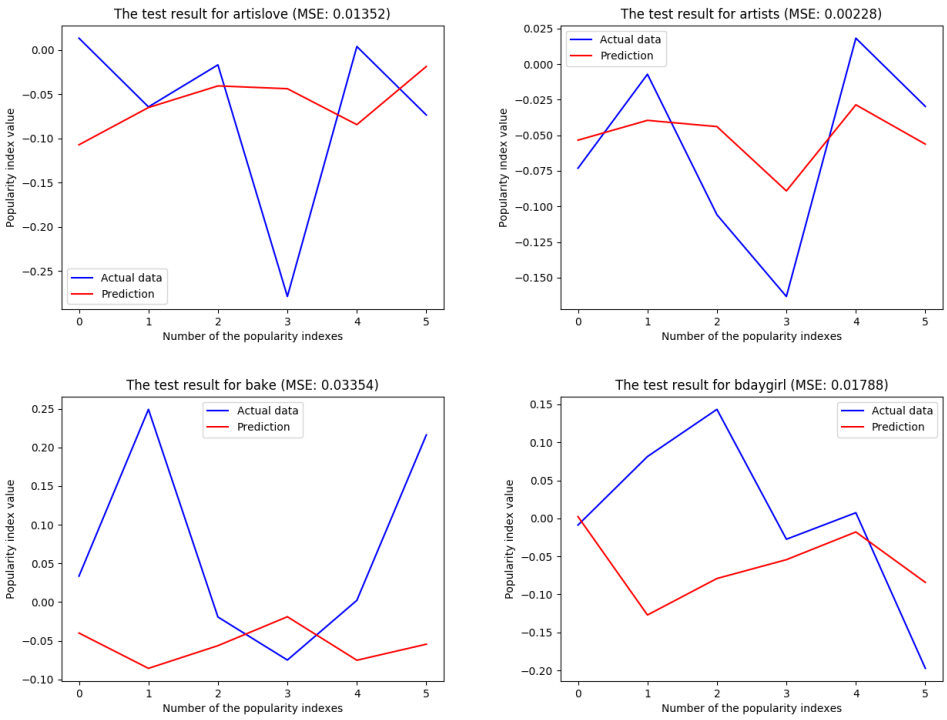

The test result for partyplanner (MSE: 0.02420)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

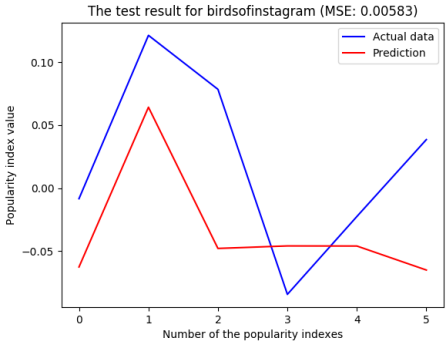

The test result for stressfree (MSE: 0.01024)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

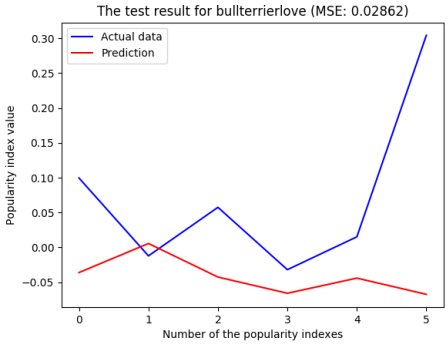

The test result for swimsuit (MSE: 0.26190)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

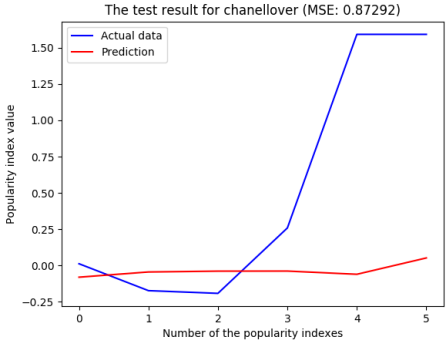


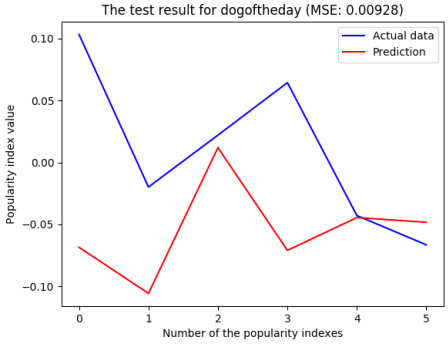
A.2.9 Batch 16, Window 5, Epochs 1000

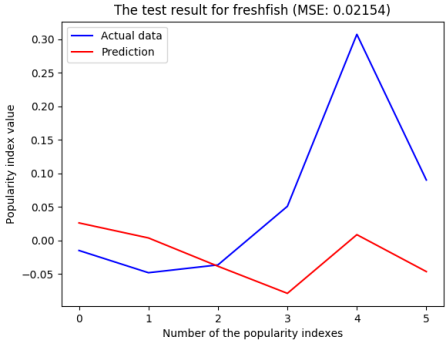


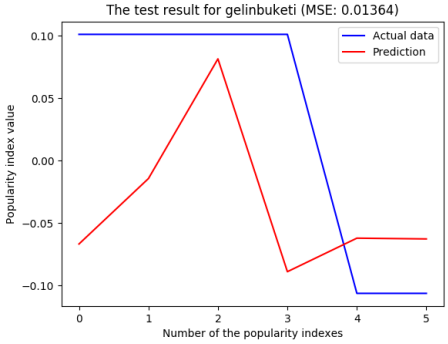

The test result for birdsofinstagram (MSE: 0.00583)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

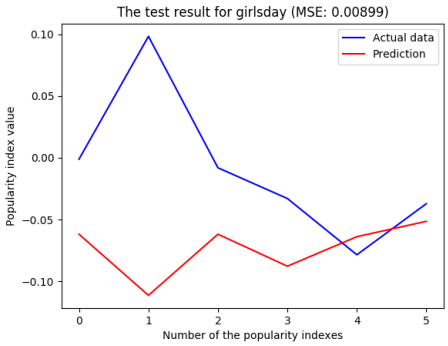

The test result for bullterrierlove (MSE: 0.02862)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

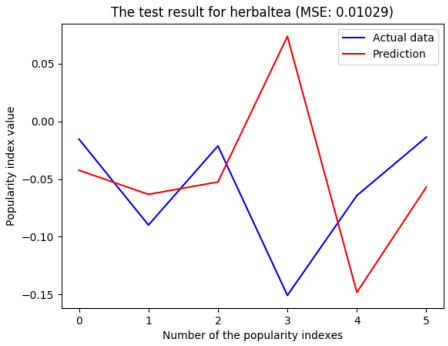

The test result for chanellover (MSE: 0.87292)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

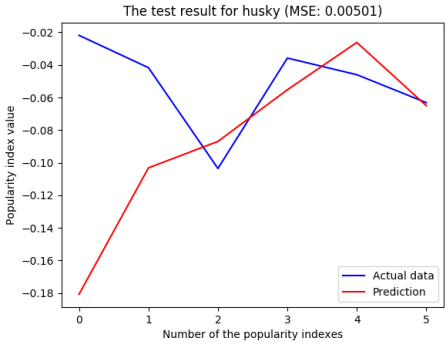

The test result for dogoftheday (MSE: 0.00928)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

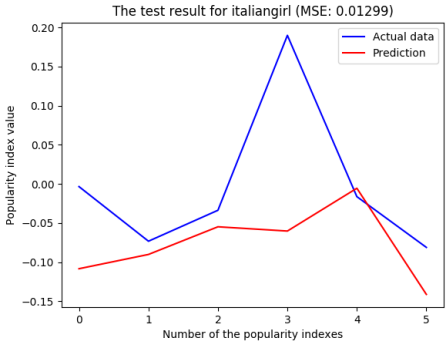

The test result for freshfish (MSE: 0.02154)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

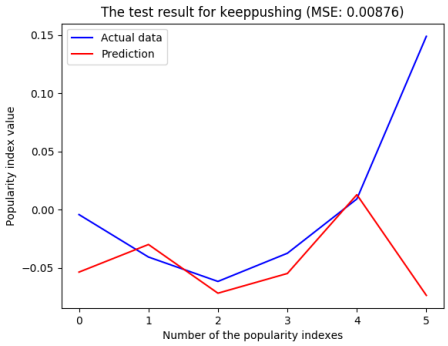

The test result for gelinbuketi (MSE: 0.01364)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

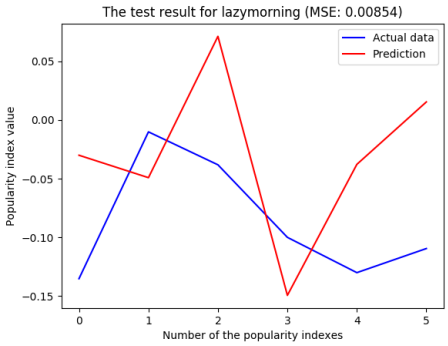

The test result for girlsday (MSE: 0.00899)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for herbaltea (MSE: 0.01029)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

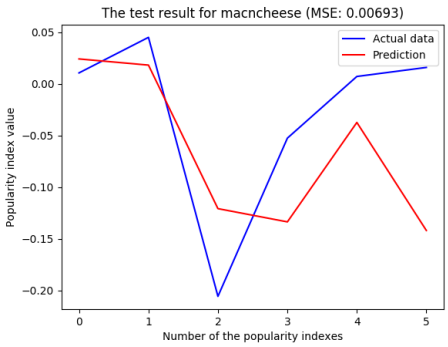

The test result for husky (MSE: 0.00501)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

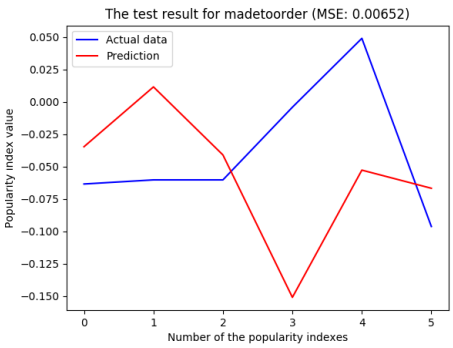

The test result for italiangirl (MSE: 0.01299)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

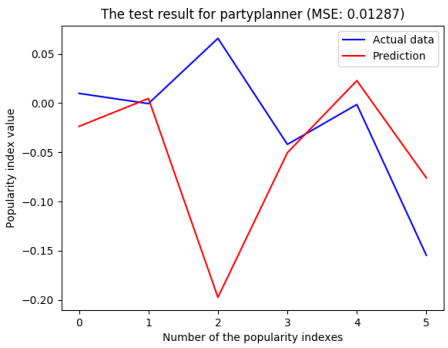

The test result for keeppushing (MSE: 0.00876)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

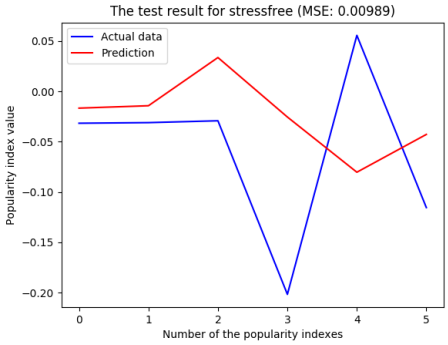

The test result for lazymorning (MSE: 0.00854)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

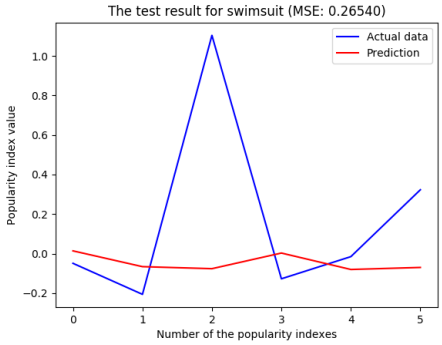

The test result for loves (MSE: 0.00500)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

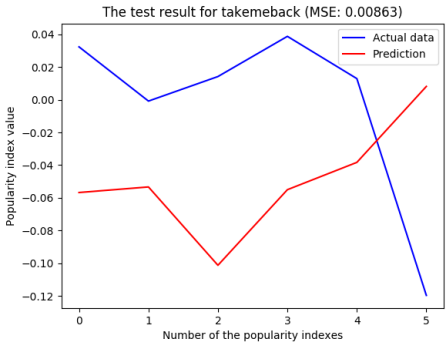

The test result for macncheese (MSE: 0.00693)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

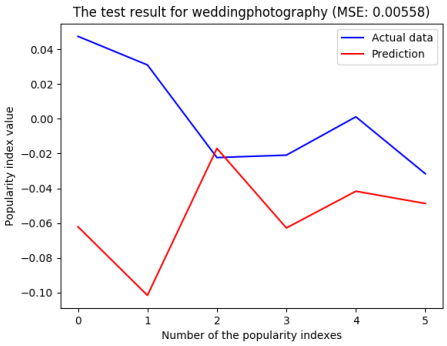

The test result for madetoorder (MSE: 0.00652)
Popularity index value
Number of the popularity indexes
Actual data
Prediction

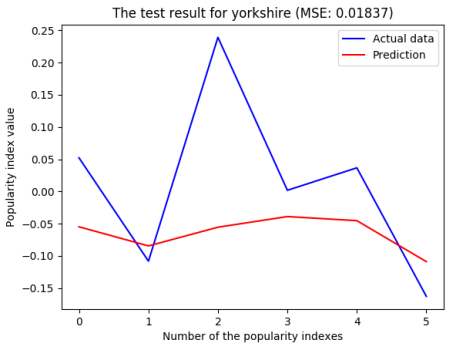

The test result for partyplanner (MSE: 0.01287)
Popularity index value
Number of the popularity indexes
Actual data
Prediction


The test result for stressfree (MSE: 0.00989)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

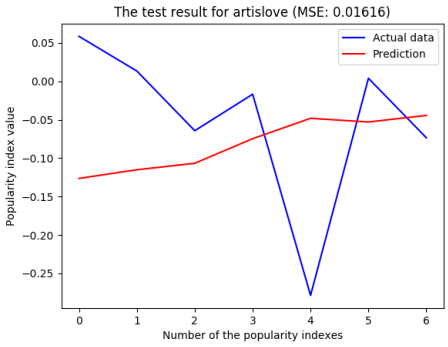

The test result for swimsuit (MSE: 0.26540)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

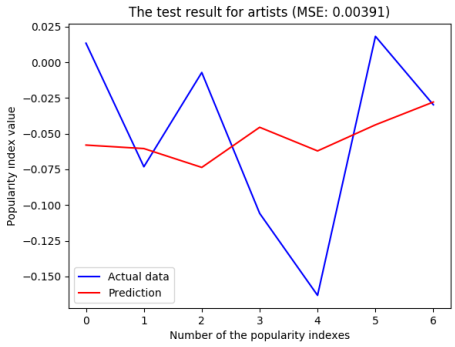

The test result for takemeback (MSE: 0.00863)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

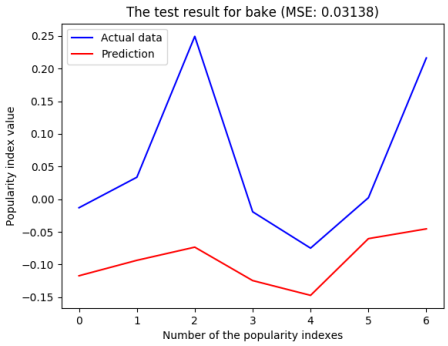

The test result for weddingphotography (MSE: 0.00558)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

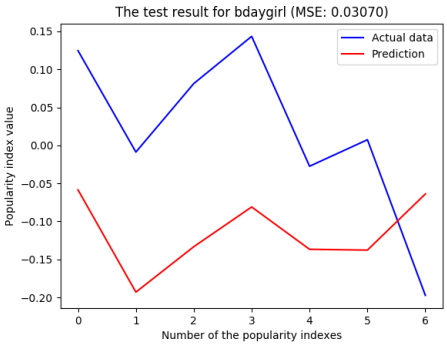

The test result for yorkshire (MSE: 0.01837)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

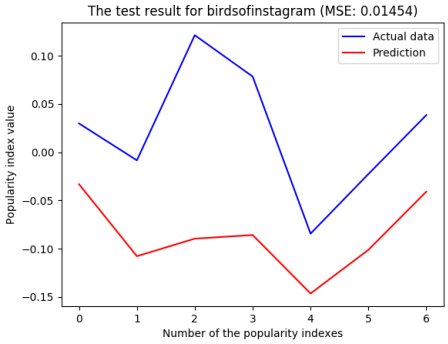
A.2.10 Batch 32, Window 3, Epochs 100

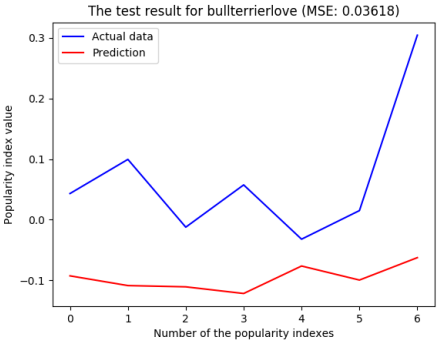


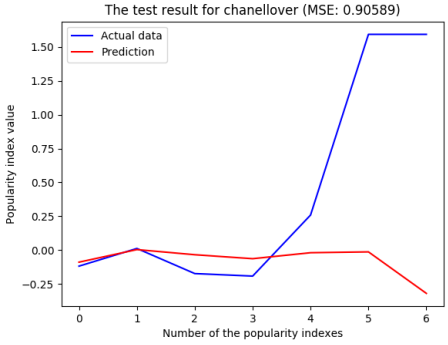


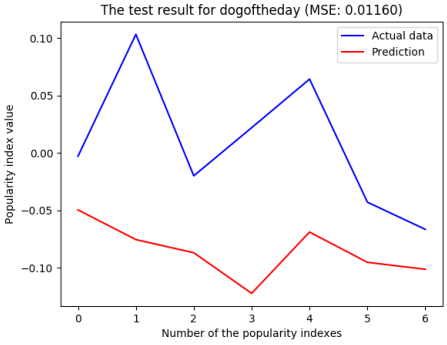


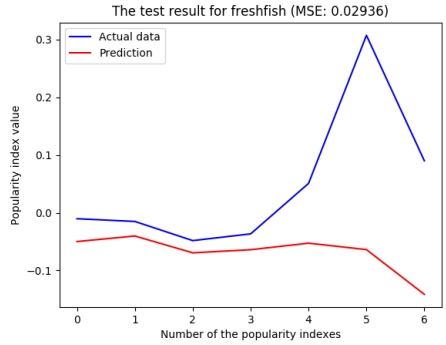


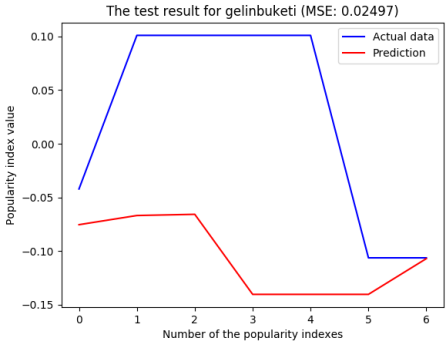


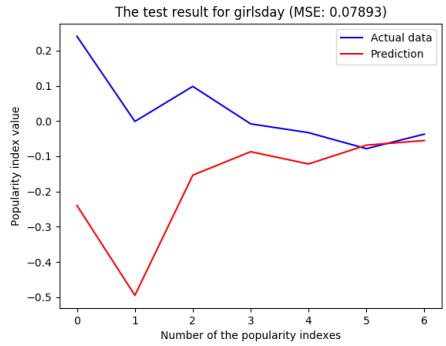


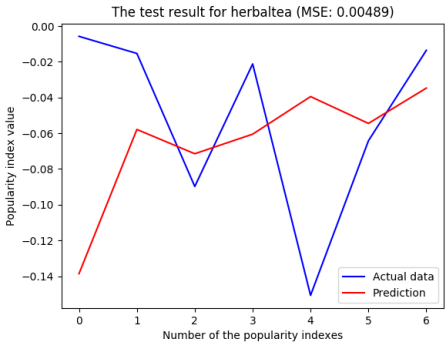


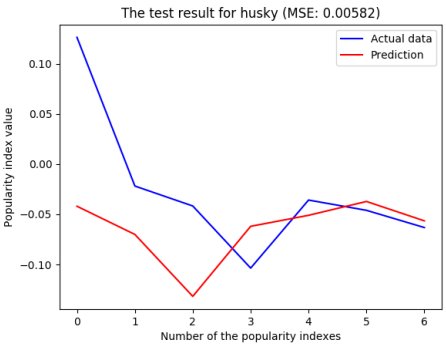


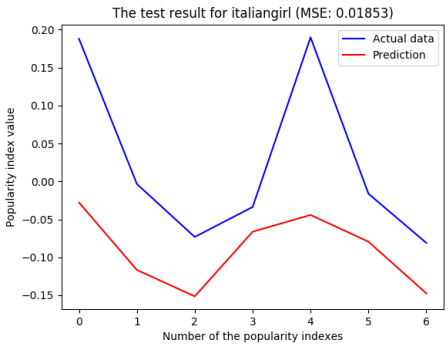

The test result for freshfish (MSE: 0.02936)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

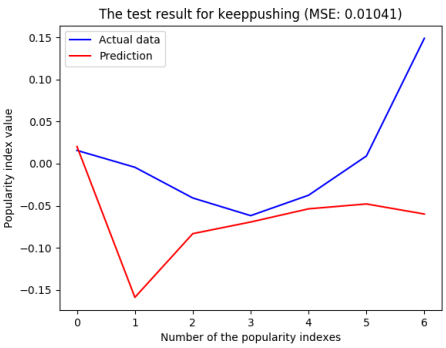

The test result for gelinbuketi (MSE: 0.02497)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

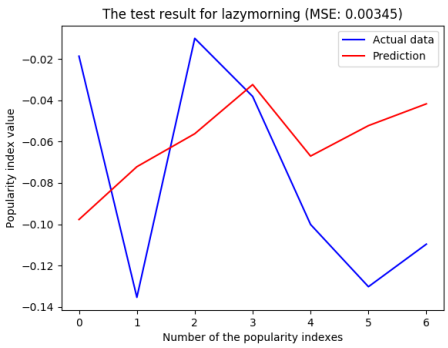

The test result for girlsday (MSE: 0.07893)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

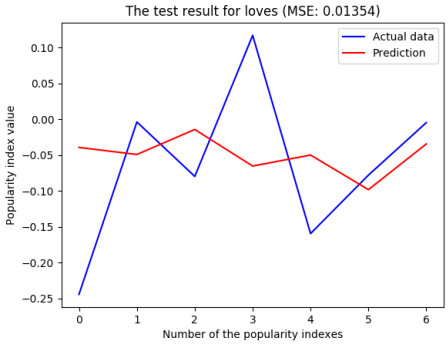

The test result for herbaltea (MSE: 0.00489)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

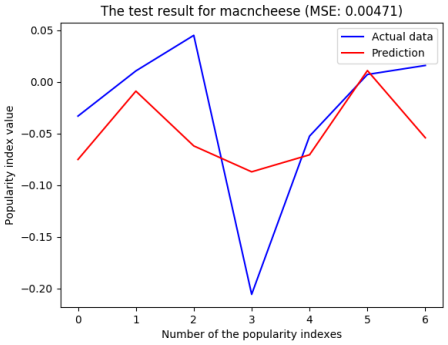

The test result for husky (MSE: 0.00582)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

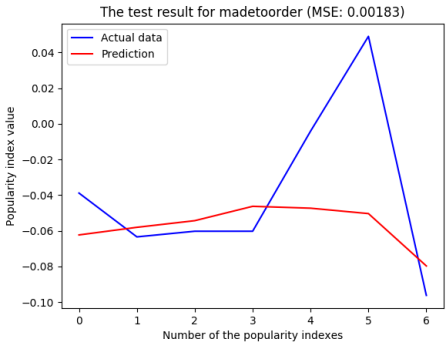

The test result for italiangirl (MSE: 0.01853)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

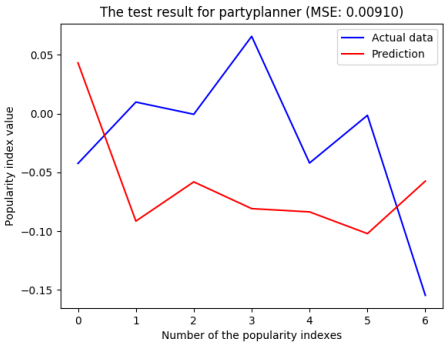

The test result for keeppushing (MSE: 0.01041)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

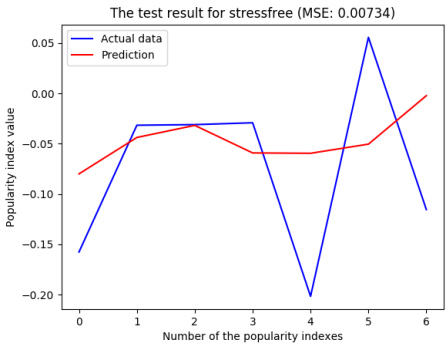

The test result for lazymorning (MSE: 0.00345)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

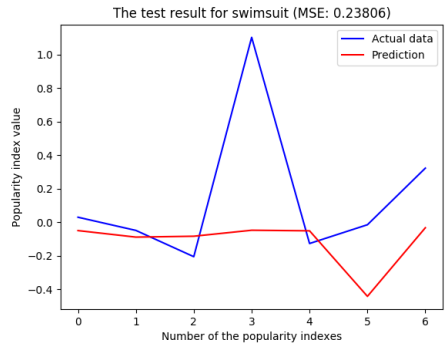

The test result for loves (MSE: 0.01354)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

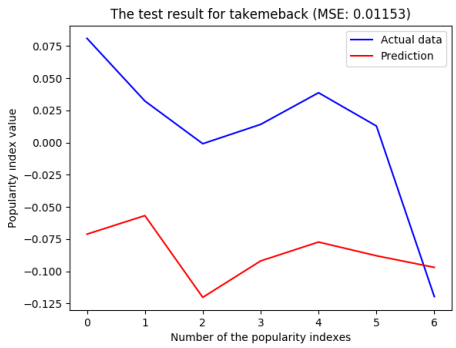

The test result for macncheese (MSE: 0.00471)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

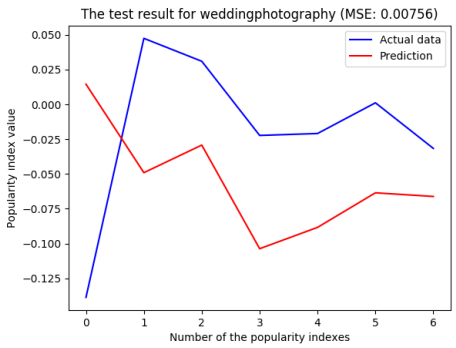

The test result for madetoorder (MSE: 0.00183)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

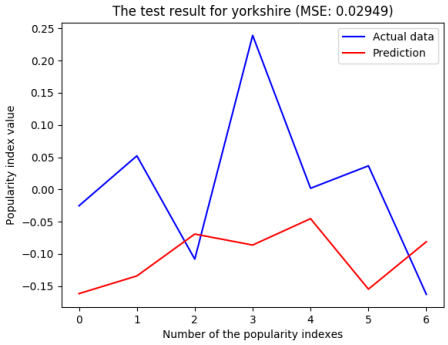

The test result for partyplanner (MSE: 0.00910)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.00734)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

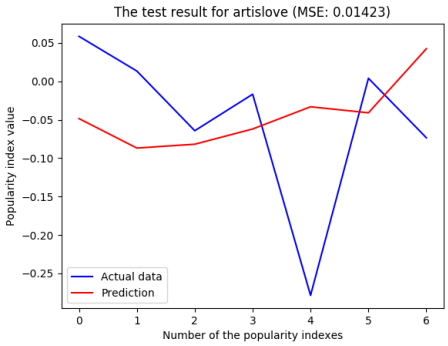

The test result for swimsuit (MSE: 0.23806)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

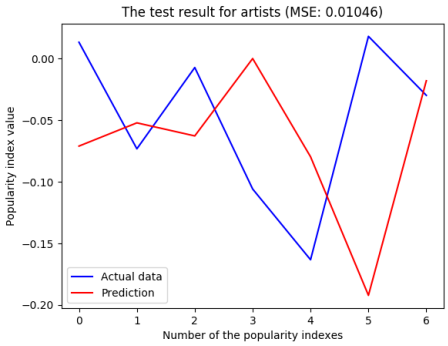

The test result for takemeback (MSE: 0.01153)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

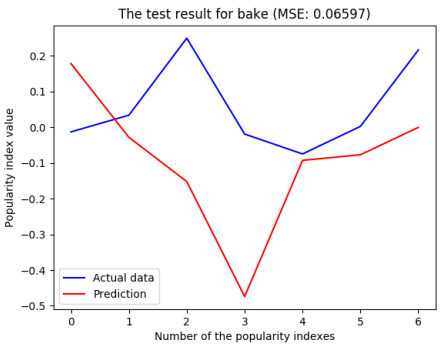

The test result for weddingphotography (MSE: 0.00756)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

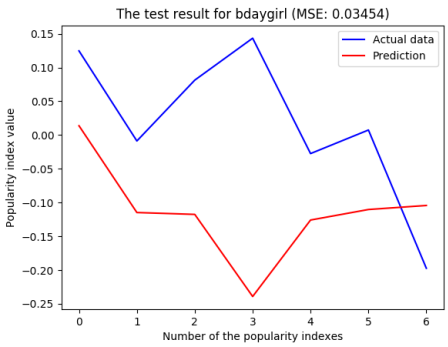


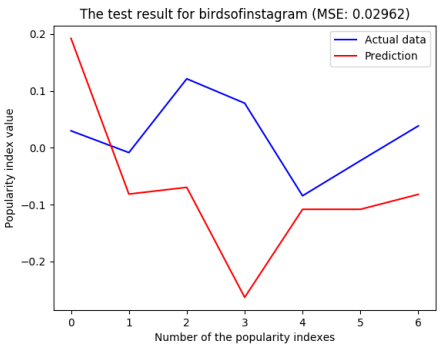
A.2.11 Batch 32, Window 3, Epochs 500

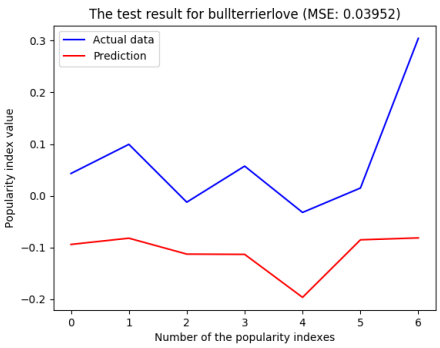




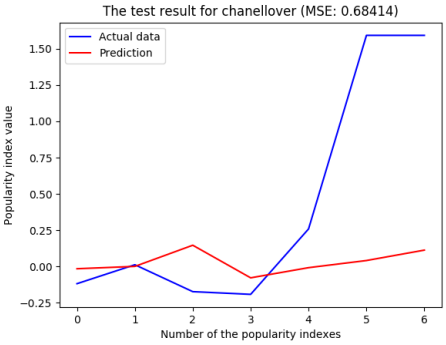


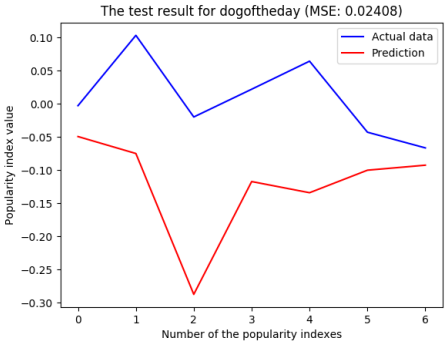


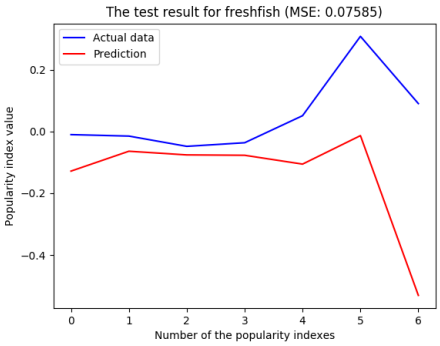


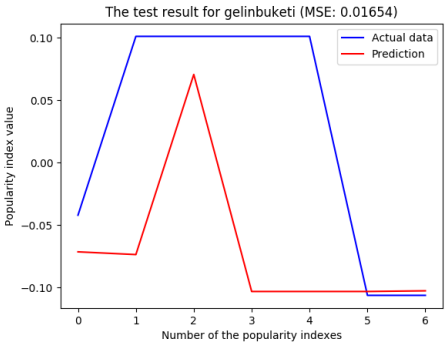


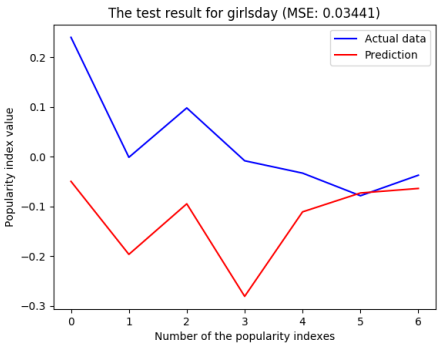
Clusters

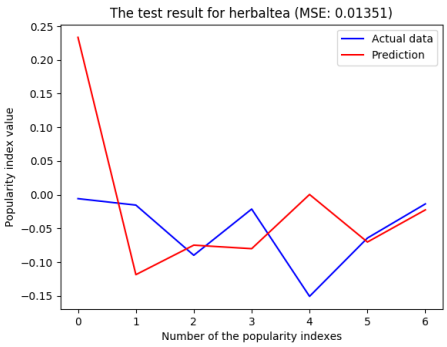

The test result for chanellover (MSE: 0.68414)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

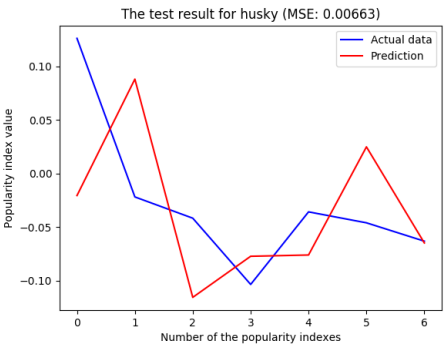

The test result for dogoftheday (MSE: 0.02408)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

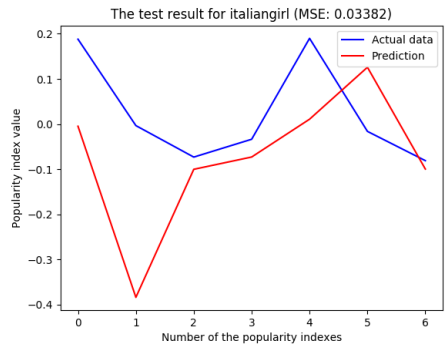

The test result for freshfish (MSE: 0.07585)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

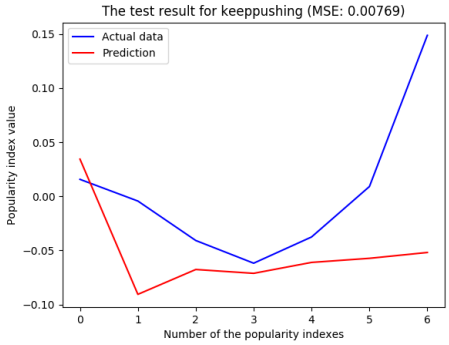

The test result for gelinbuketi (MSE: 0.01654)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

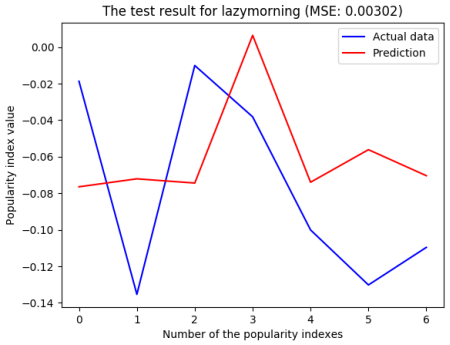

The test result for girlsday (MSE: 0.03441)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

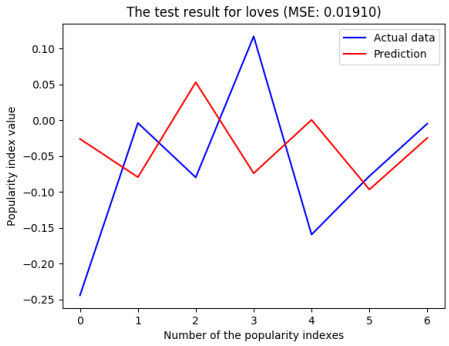

The test result for herbaltea (MSE: 0.01351)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

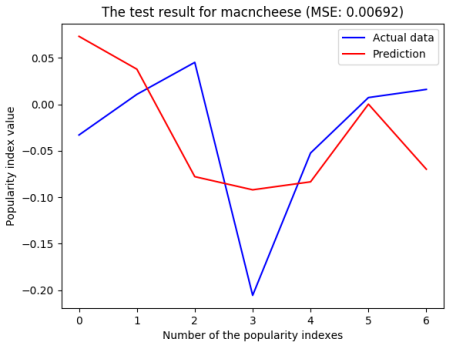

The test result for husky (MSE: 0.00663)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

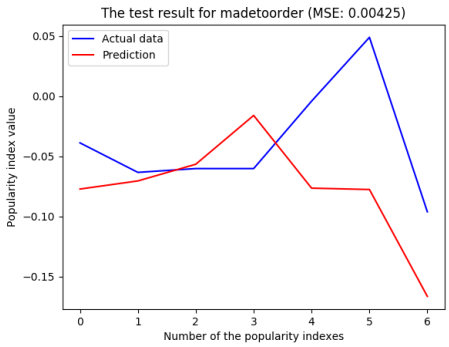

The test result for italiangirl (MSE: 0.03382)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

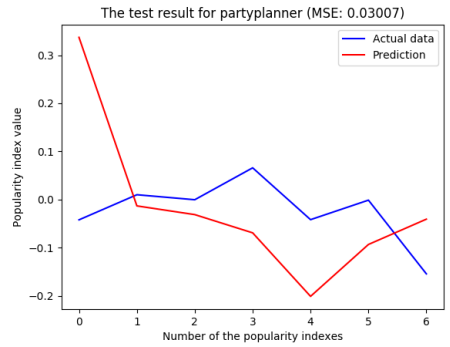

The test result for keeppushing (MSE: 0.00769)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

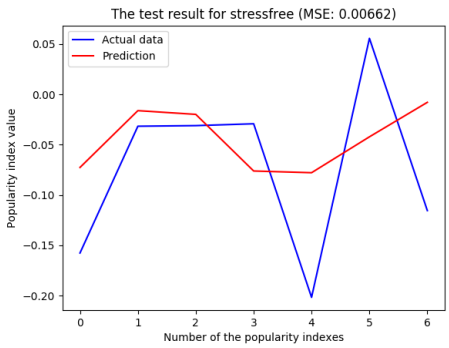

The test result for lazymorning (MSE: 0.00302)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

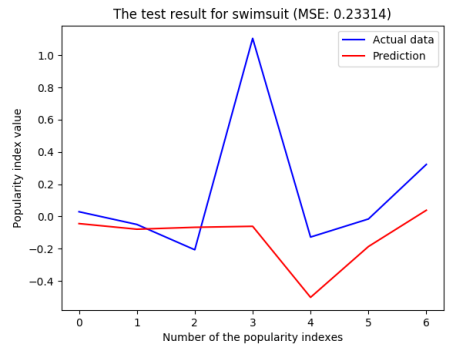

The test result for loves (MSE: 0.01910)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

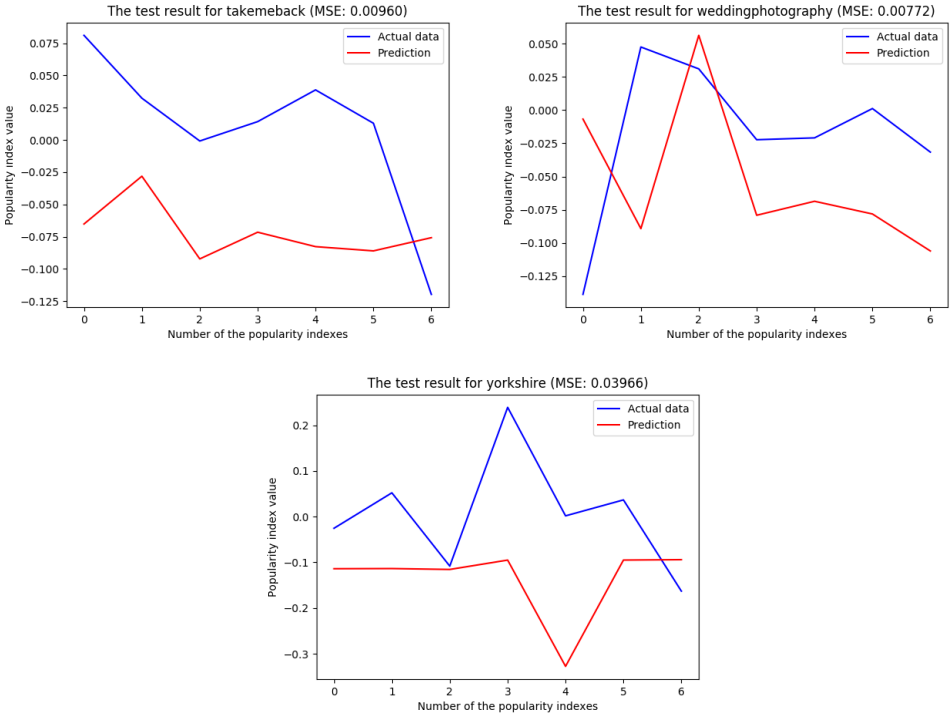

The test result for macncheese (MSE: 0.00692)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00425)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

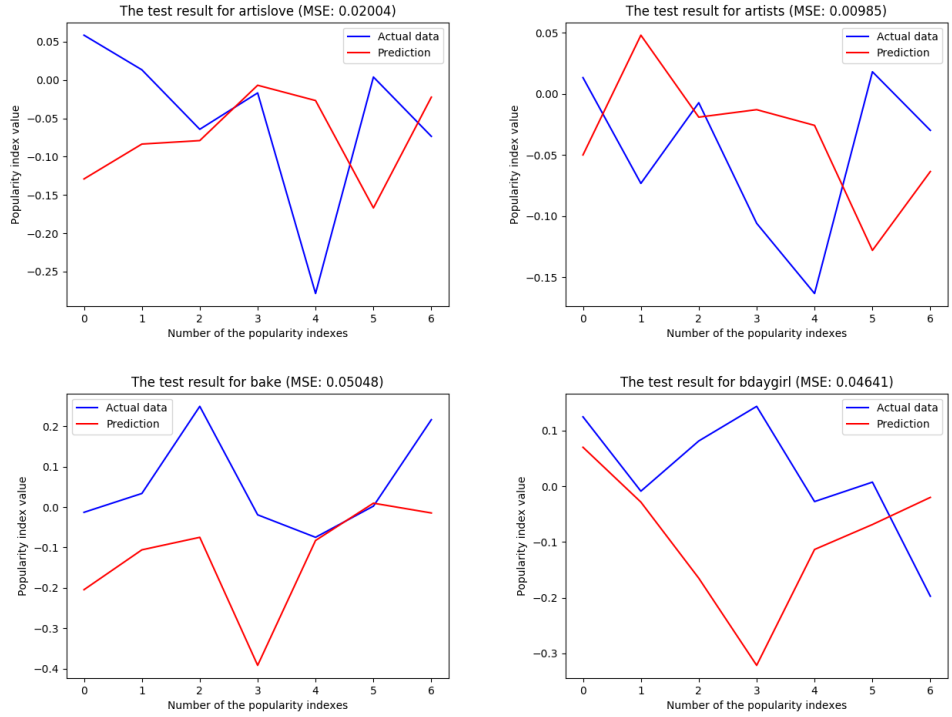

The test result for partyplanner (MSE: 0.03007)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

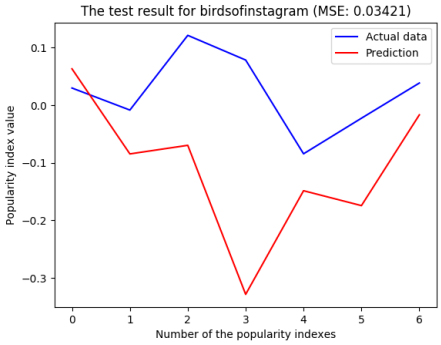

The test result for stressfree (MSE: 0.00662)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

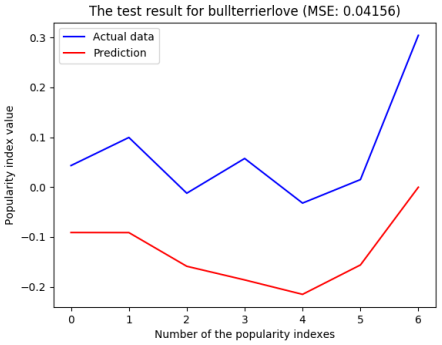

The test result for swimsuit (MSE: 0.23314)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

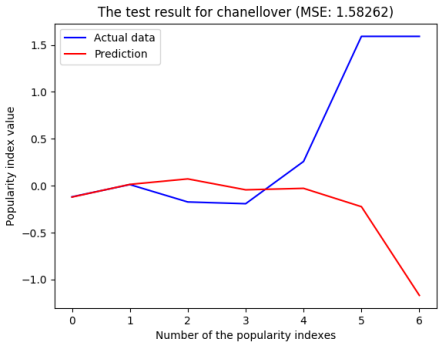


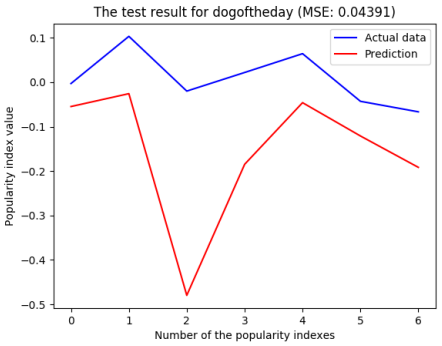
A.2.12 Batch 32, Window 3, Epochs 1000

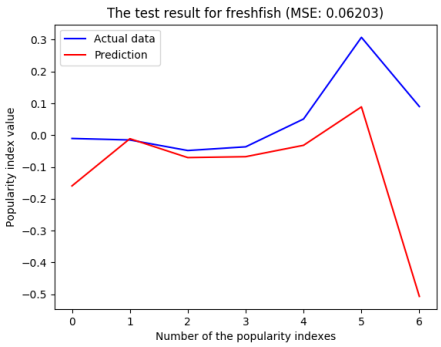


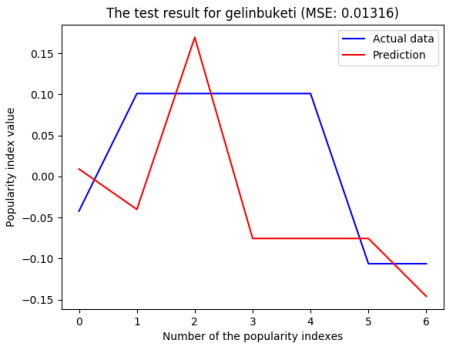

The test result for birdsofinstagram (MSE: 0.03421)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

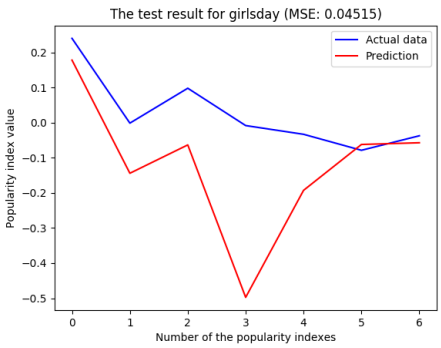

The test result for bullterrierlove (MSE: 0.04156)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

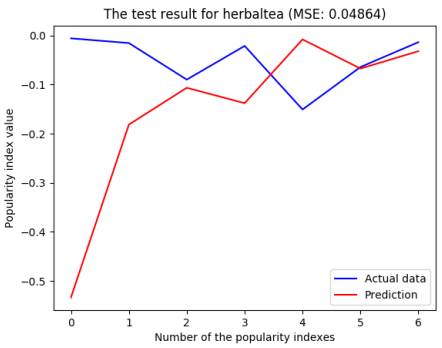

The test result for chanellover (MSE: 1.58262)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for dogoftheday (MSE: 0.04391)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

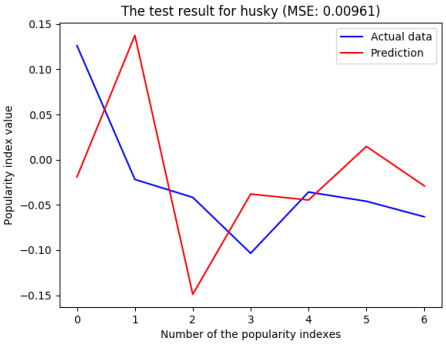

The test result for freshfish (MSE: 0.06203)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

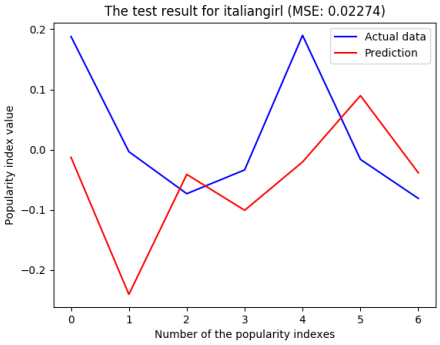

The test result for gelinbuketi (MSE: 0.01316)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

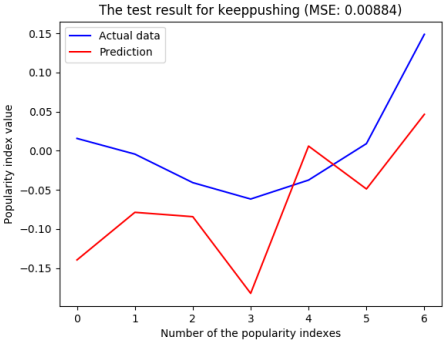

The test result for girlsday (MSE: 0.04515)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

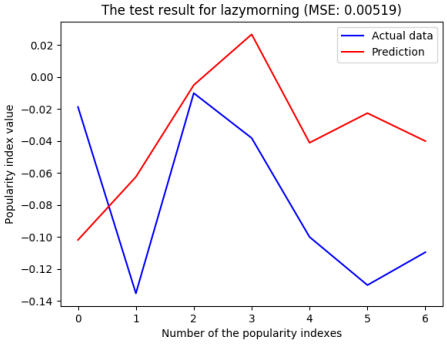

The test result for herbaltea (MSE: 0.04864)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

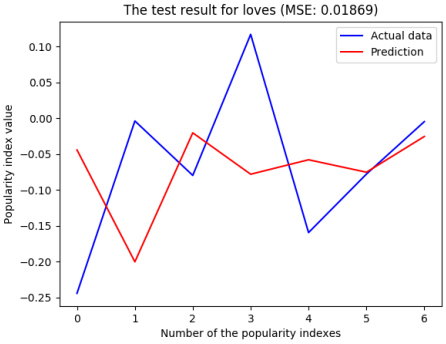
Clusters

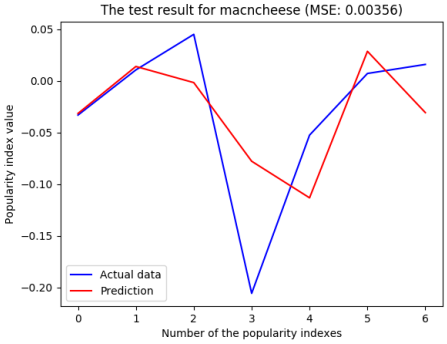

The test result for husky (MSE: 0.00961)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

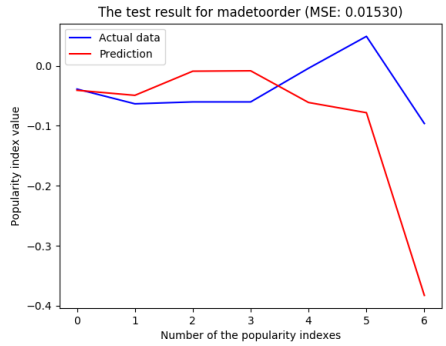

The test result for italiangirl (MSE: 0.02274)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

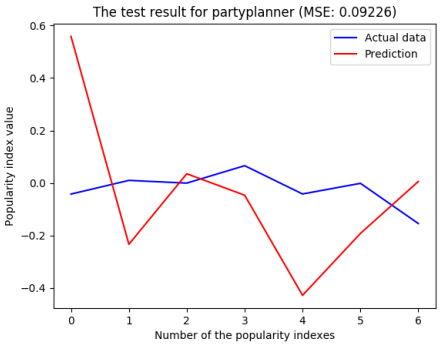

The test result for keeppushing (MSE: 0.00884)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

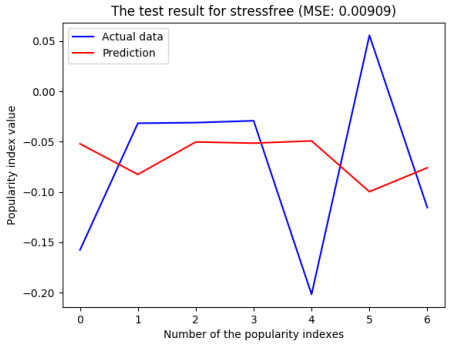

The test result for lazymorning (MSE: 0.00519)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

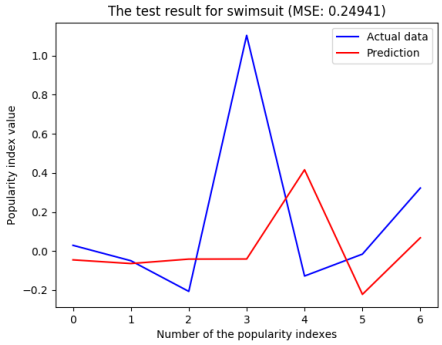

The test result for loves (MSE: 0.01869)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

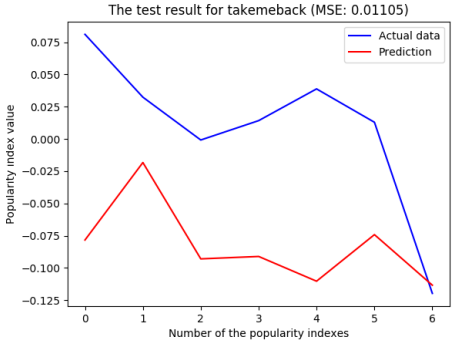

The test result for macncheese (MSE: 0.00356)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

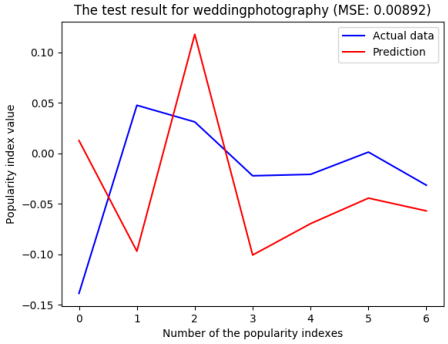

The test result for madetoorder (MSE: 0.01530)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

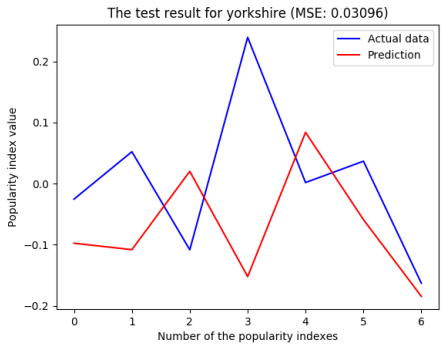

The test result for partyplanner (MSE: 0.09226)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.00909)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

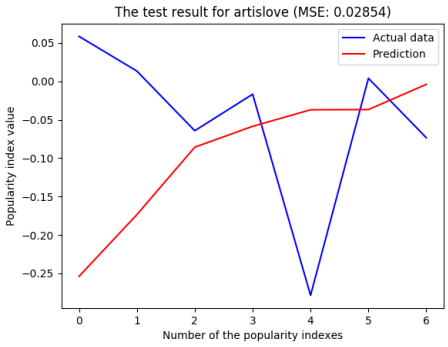

The test result for swimsuit (MSE: 0.24941)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

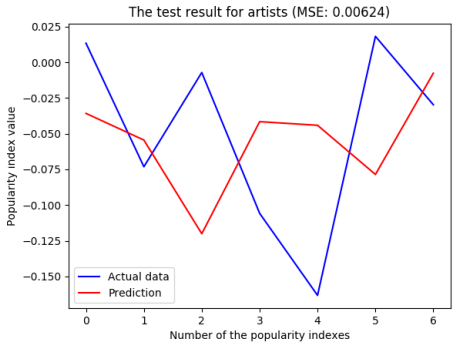

The test result for takemeback (MSE: 0.01105)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

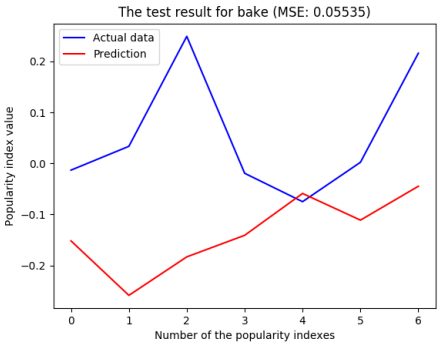

The test result for weddingphotography (MSE: 0.00892)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

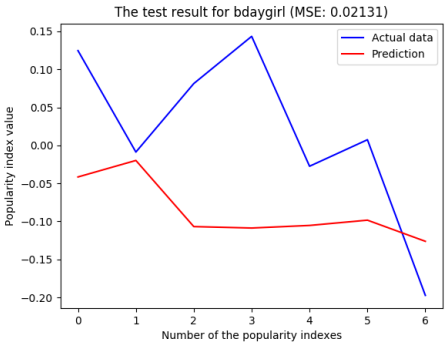

The test result for yorkshire (MSE: 0.03096)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

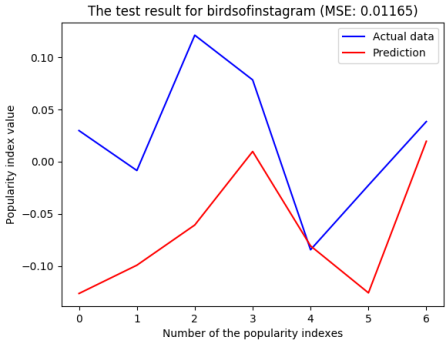
A.2.13 Batch 32, Window 4, Epochs 100

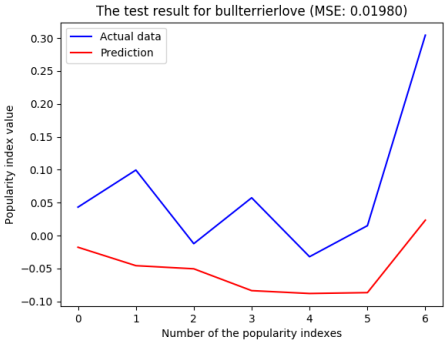


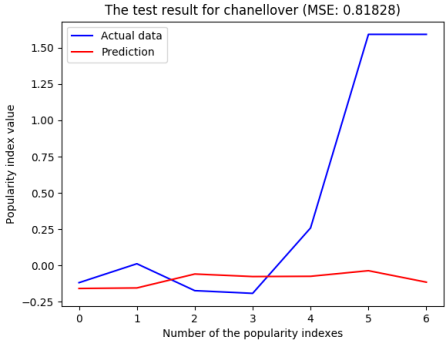


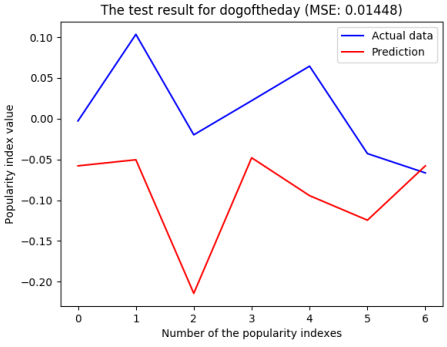


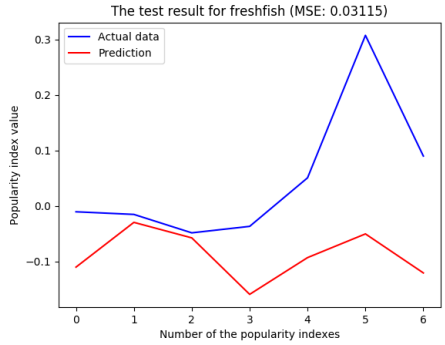


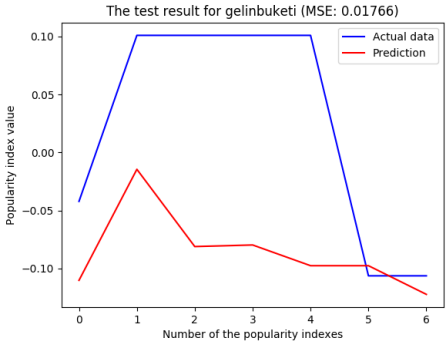


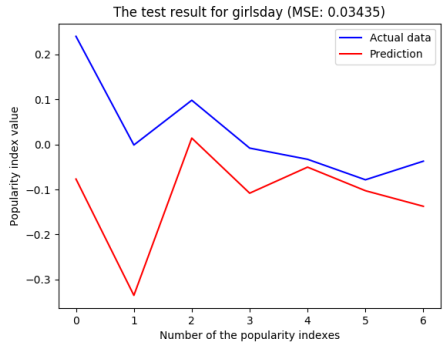


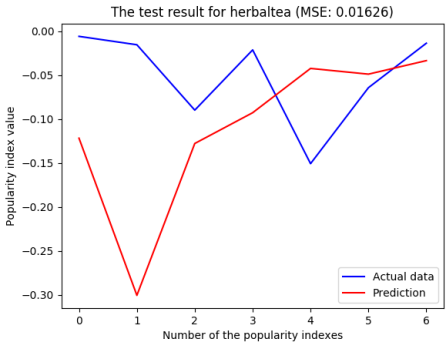


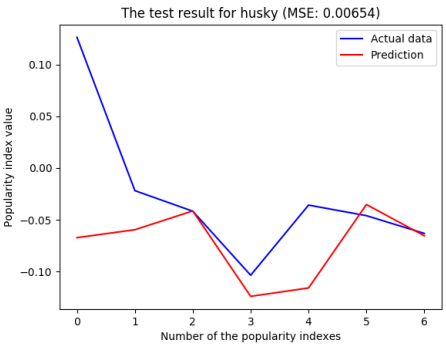


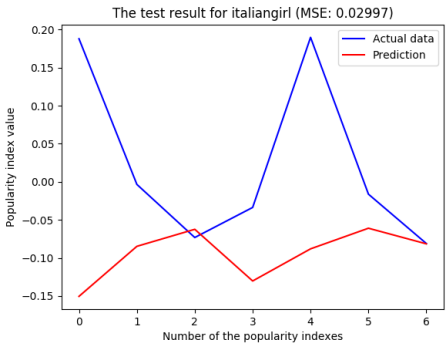

The test result for freshfish (MSE: 0.03115)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

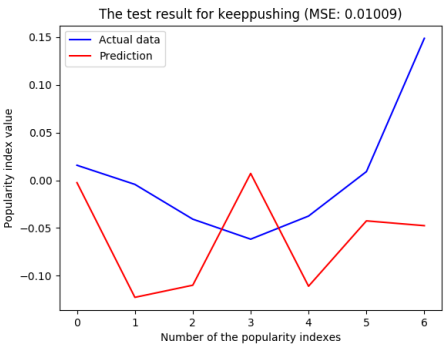

The test result for gelinbuketi (MSE: 0.01766)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

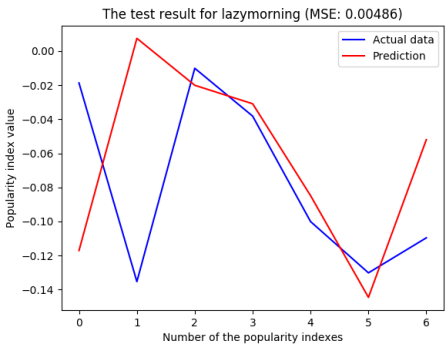

The test result for girlsday (MSE: 0.03435)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

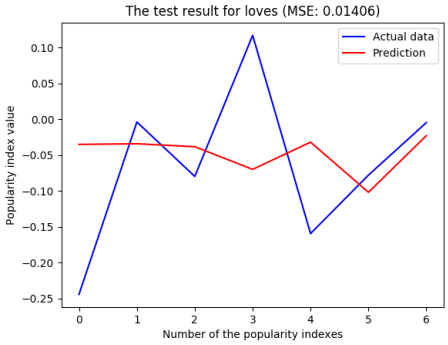

The test result for herbaltea (MSE: 0.01626)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

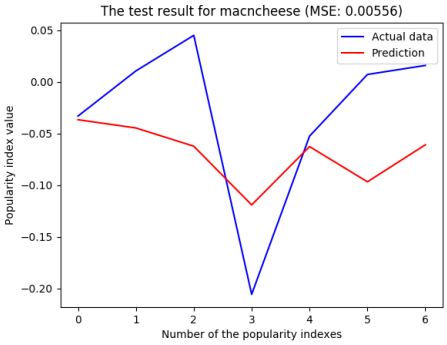

The test result for husky (MSE: 0.00654)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

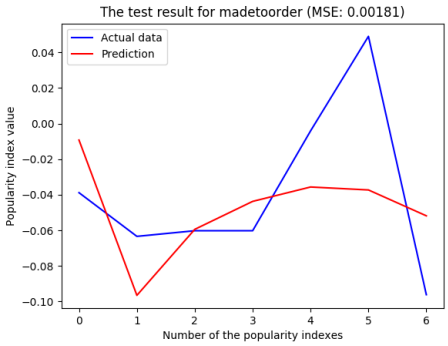

The test result for italiangirl (MSE: 0.02997)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

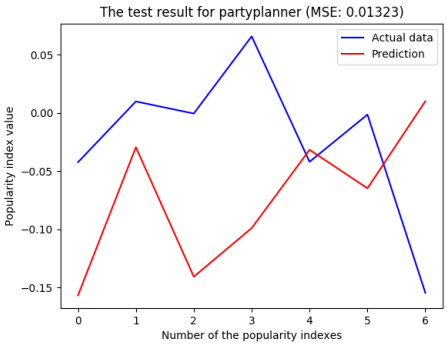

The test result for keeppushing (MSE: 0.01009)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

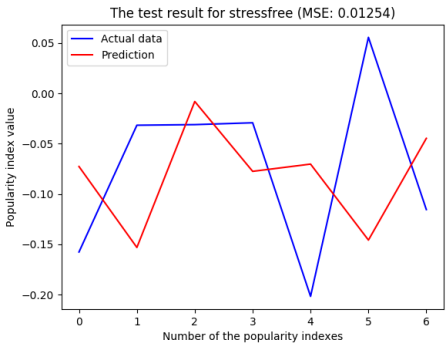

The test result for lazymorning (MSE: 0.00486)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

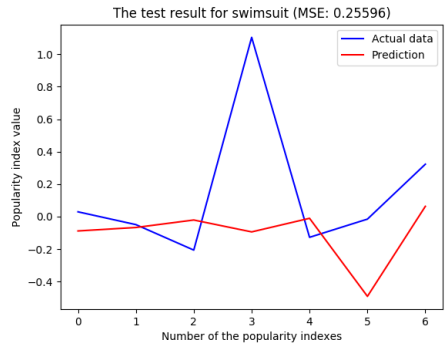

The test result for loves (MSE: 0.01406)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

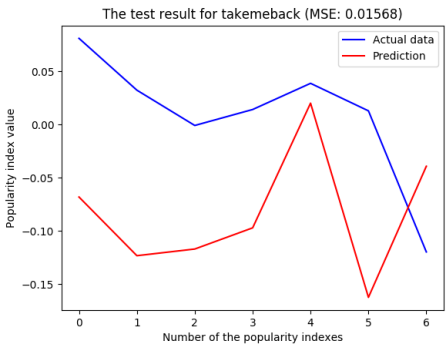

The test result for macncheese (MSE: 0.00556)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

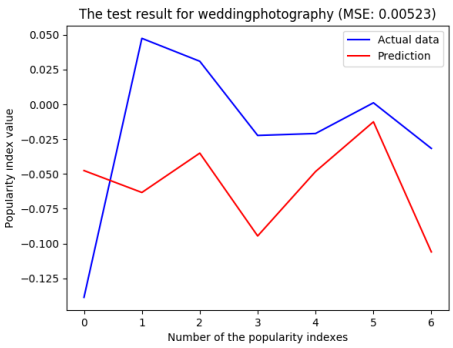

The test result for madetoorder (MSE: 0.00181)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

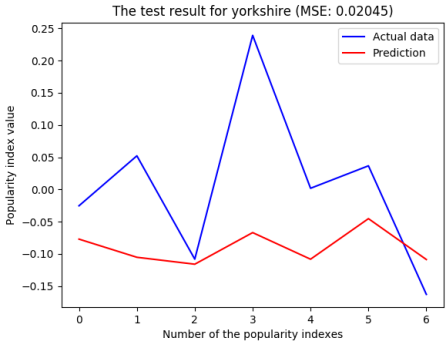

The test result for partyplanner (MSE: 0.01323)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.01254)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

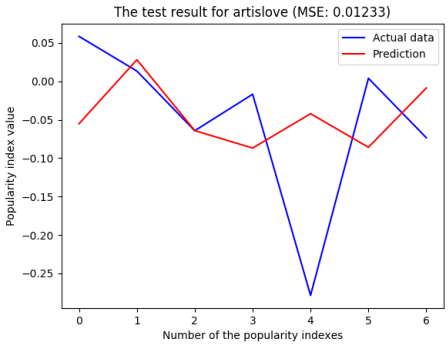

The test result for swimsuit (MSE: 0.25596)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

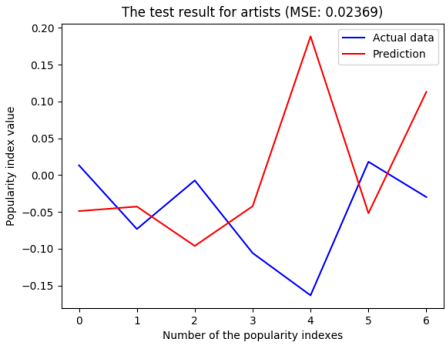

The test result for takemeback (MSE: 0.01568)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

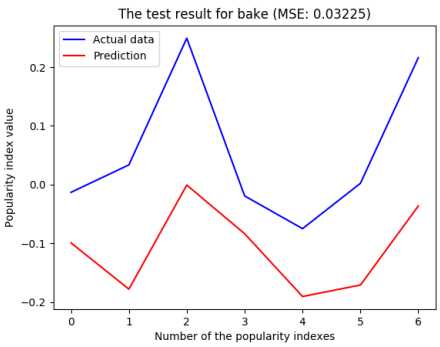

The test result for weddingphotography (MSE: 0.00523)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

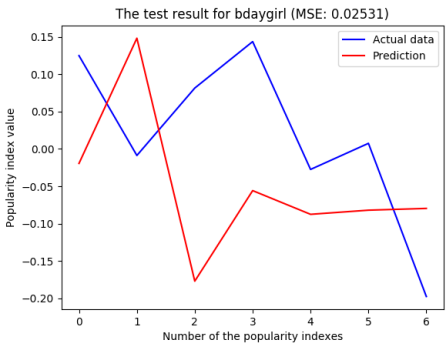


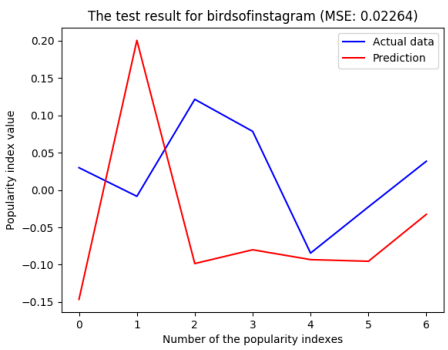
A.2.14 Batch 32, Window 4, Epochs 500

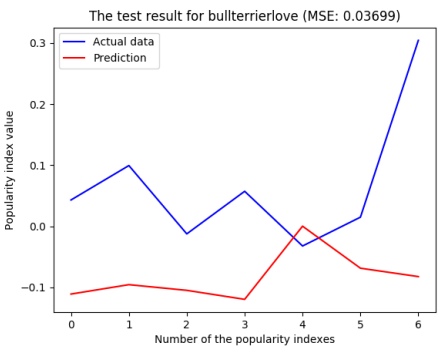


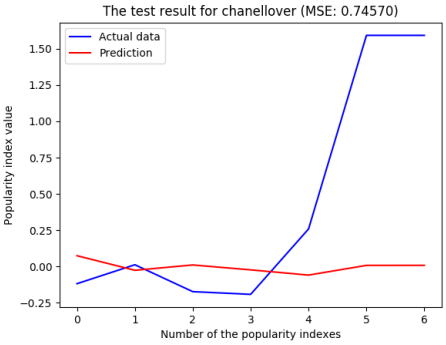


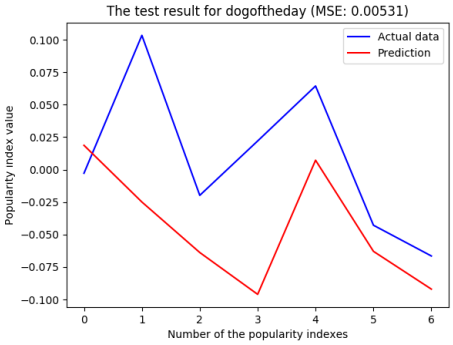


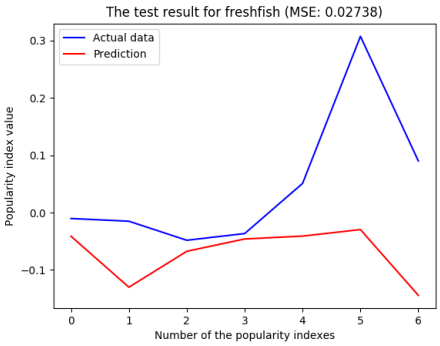


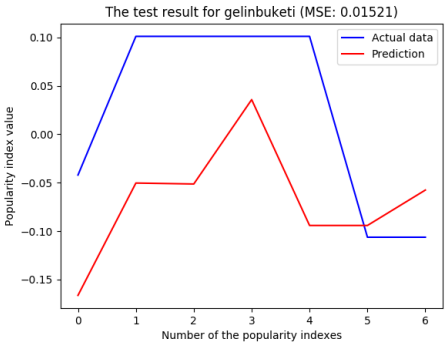


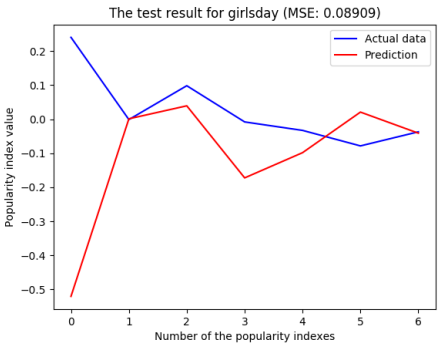


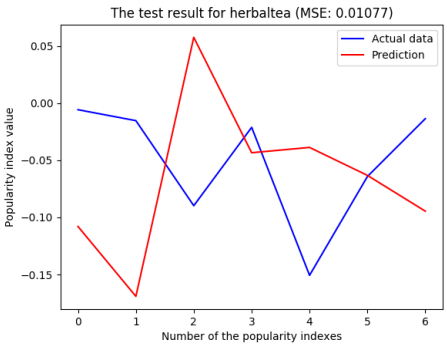

The test result for chanellover (MSE: 0.74570)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

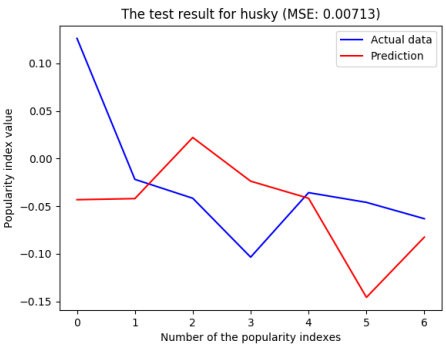

The test result for dogoftheday (MSE: 0.00531)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

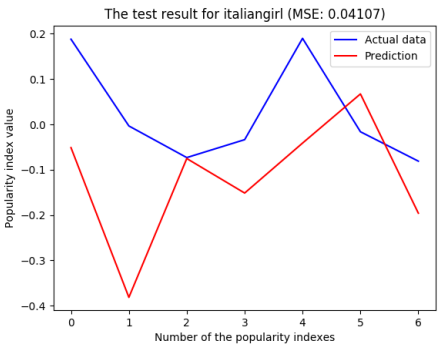

The test result for freshfish (MSE: 0.02738)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

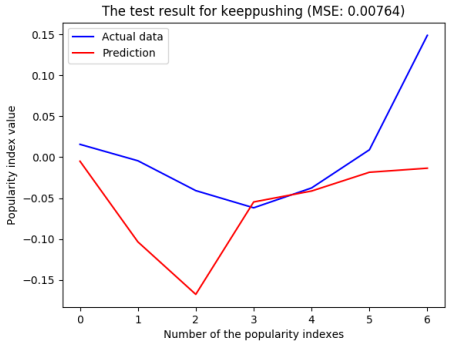

The test result for gelinbuketi (MSE: 0.01521)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

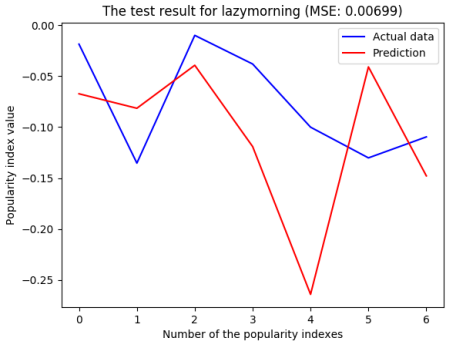

The test result for girlsday (MSE: 0.08909)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

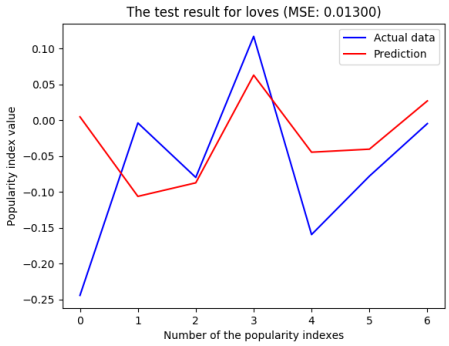

The test result for herbaltea (MSE: 0.01077)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

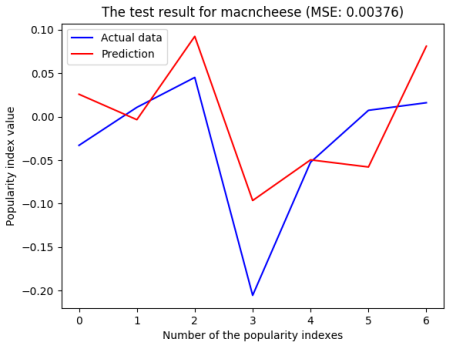

The test result for husky (MSE: 0.00713)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

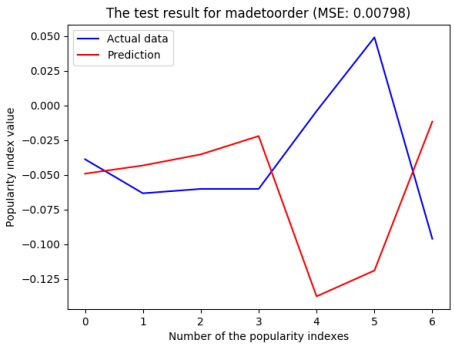

The test result for italiangirl (MSE: 0.04107)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

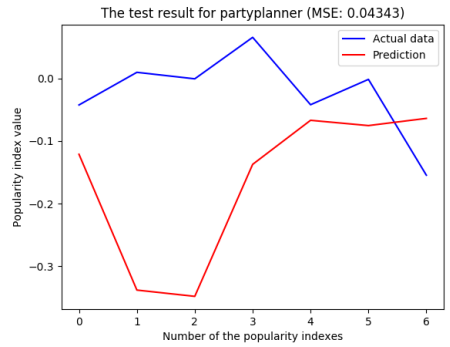

The test result for keeppushing (MSE: 0.00764)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

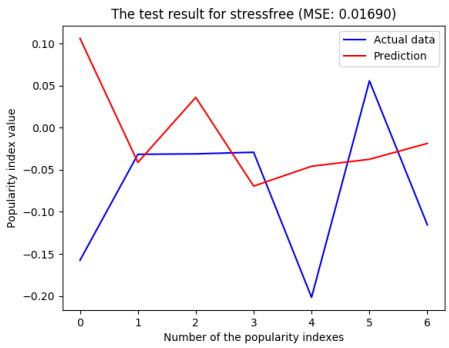

The test result for lazymorning (MSE: 0.00699)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

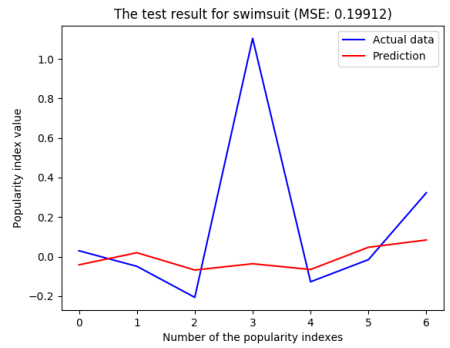

The test result for loves (MSE: 0.01300)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

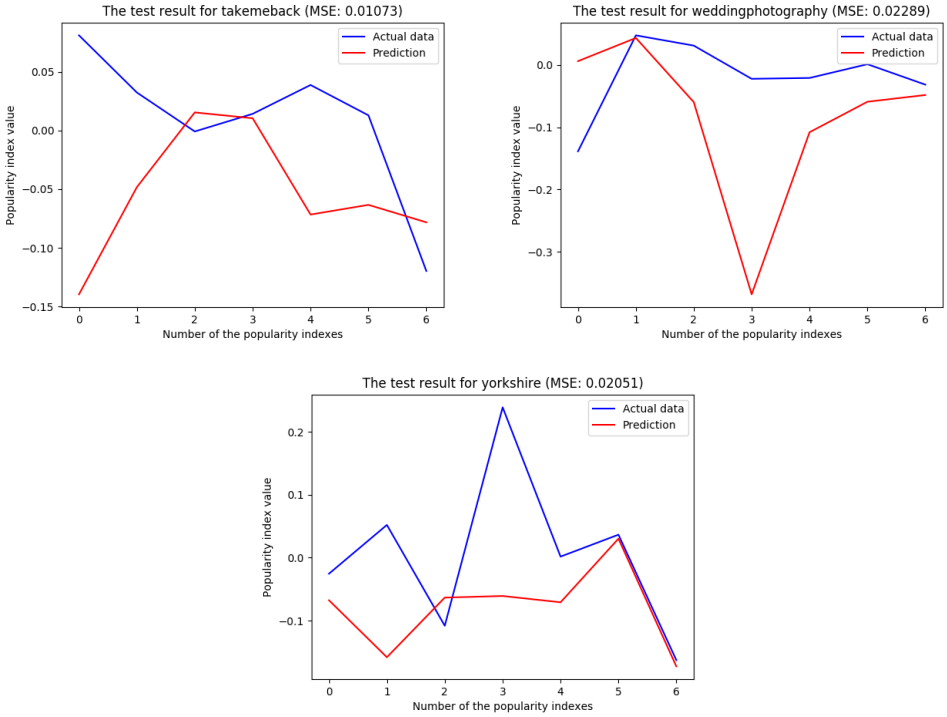

The test result for macncheese (MSE: 0.00376)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00798)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

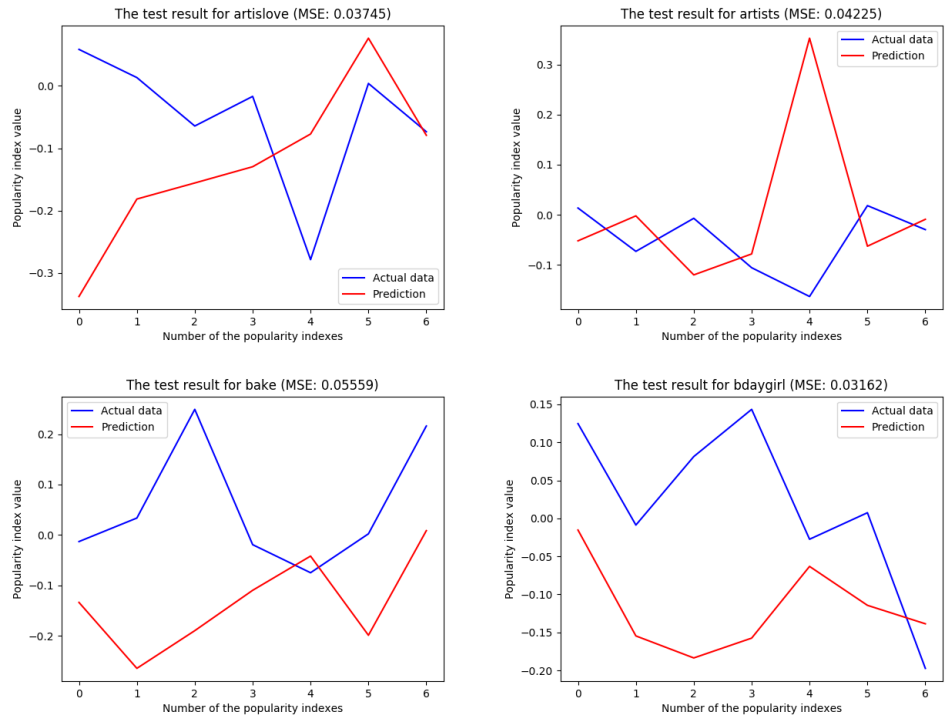

The test result for partyplanner (MSE: 0.04343)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

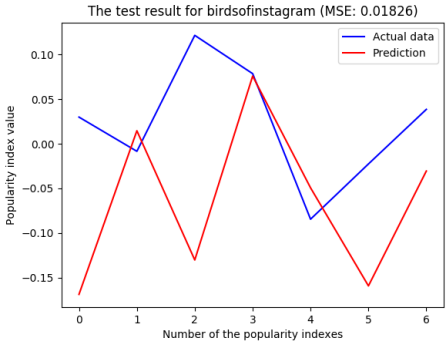

The test result for stressfree (MSE: 0.01690)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

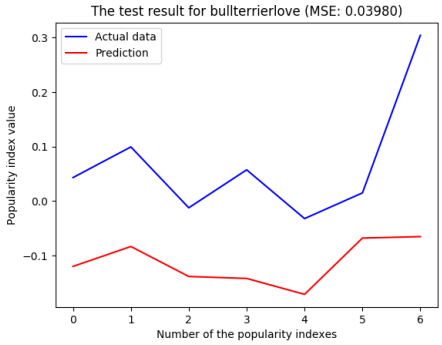

The test result for swimsuit (MSE: 0.19912)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

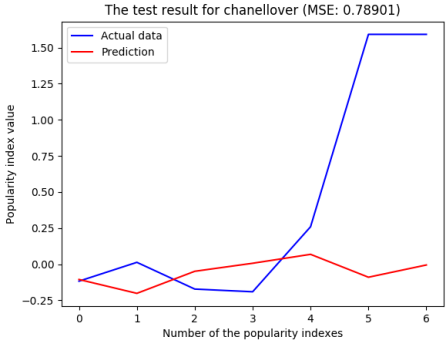


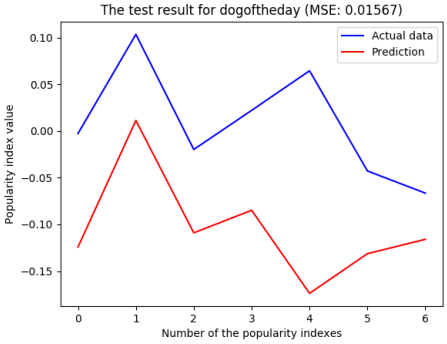
A.2.15 Batch 32, Window 4, Epochs 1000

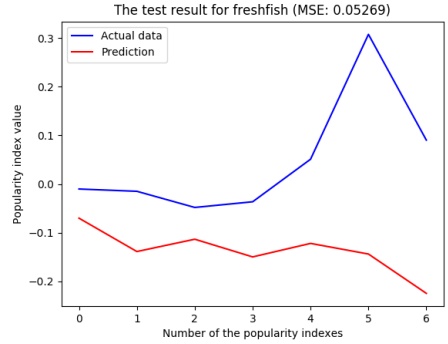


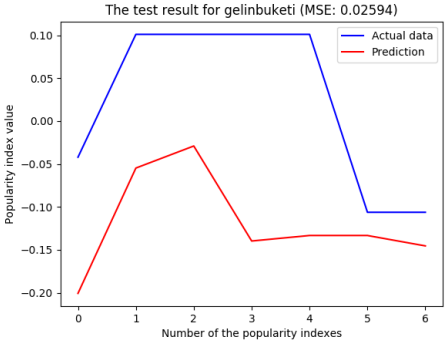

The test result for birdsofinstagram (MSE: 0.01826)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

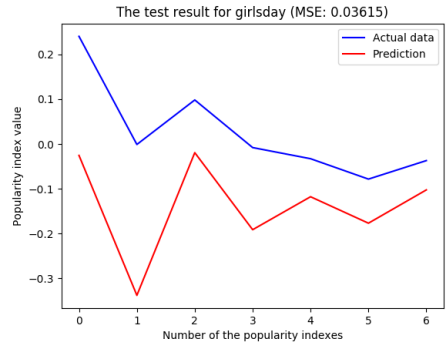

The test result for bullterrierlove (MSE: 0.03980)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

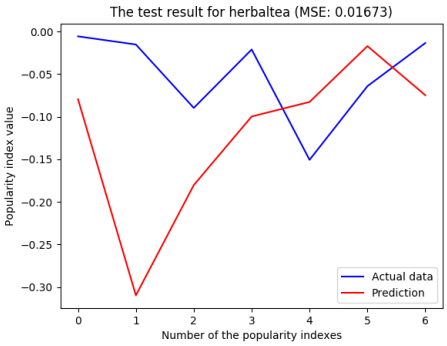

The test result for chanellover (MSE: 0.78901)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

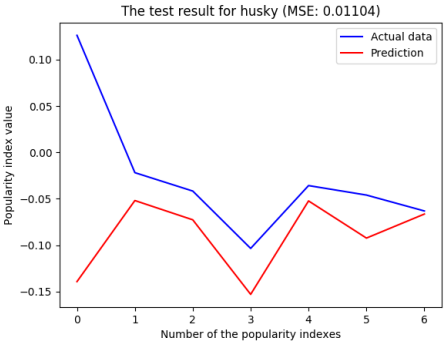

The test result for dogoftheday (MSE: 0.01567)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

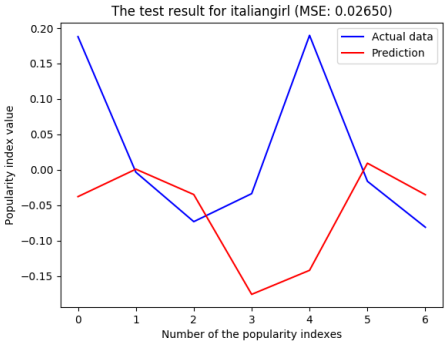

The test result for freshfish (MSE: 0.05269)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

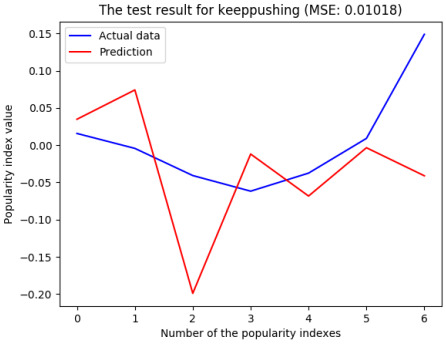

The test result for gelinbuketi (MSE: 0.02594)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

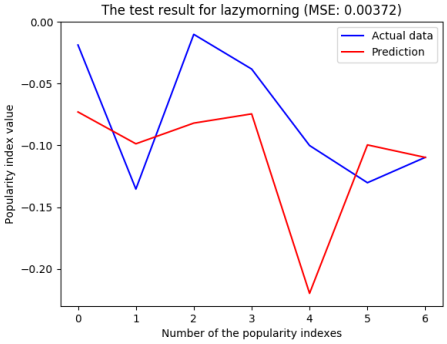

The test result for girlsday (MSE: 0.03615)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

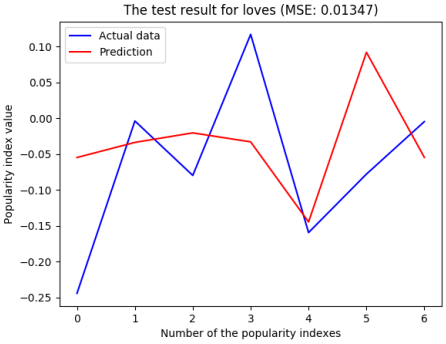

The test result for herbaltea (MSE: 0.01673)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

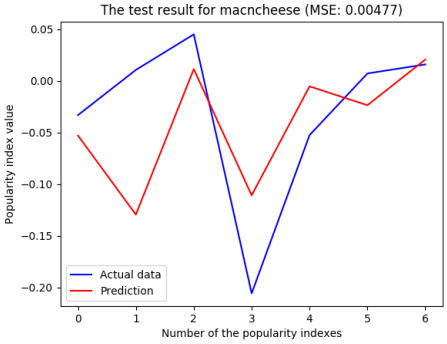

The test result for husky (MSE: 0.01104)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

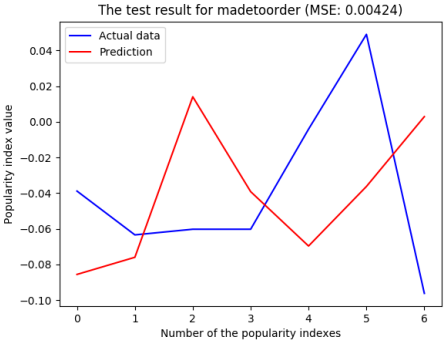

The test result for italiangirl (MSE: 0.02650)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

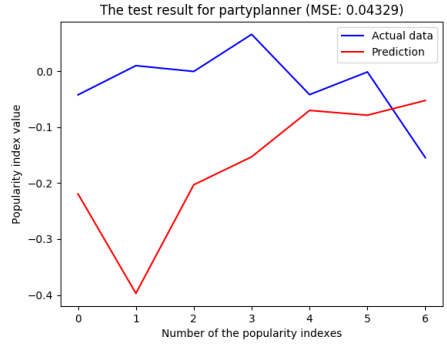

The test result for keeppushing (MSE: 0.01018)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

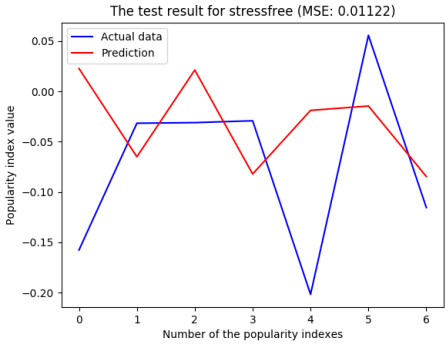

The test result for lazymorning (MSE: 0.00372)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

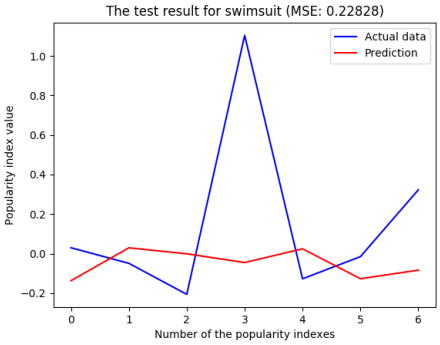

The test result for loves (MSE: 0.01347)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

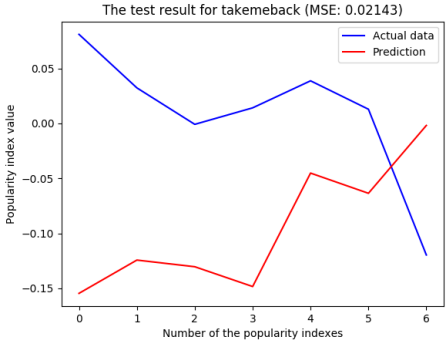

The test result for macncheese (MSE: 0.00477)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00424)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

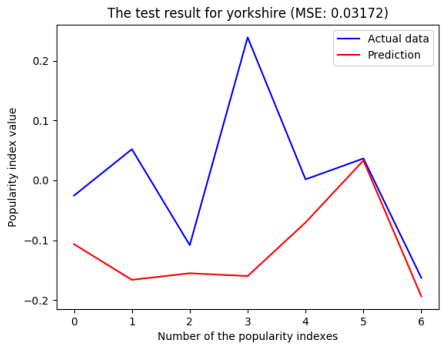

The test result for partyplanner (MSE: 0.04329)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.01122)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

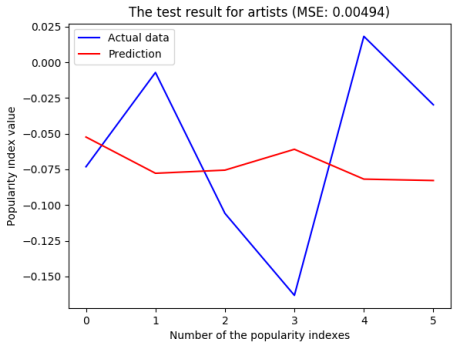

The test result for swimsuit (MSE: 0.22828)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

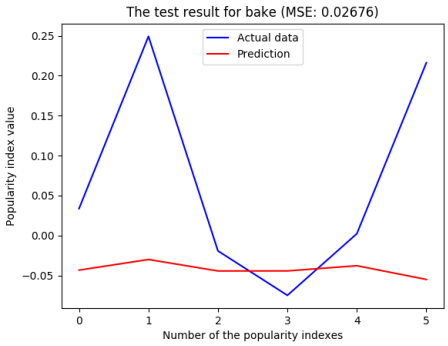

The test result for takemeback (MSE: 0.02143)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

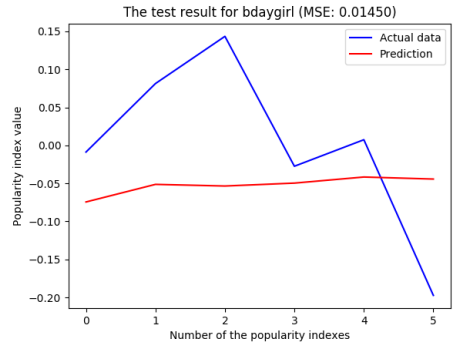

The test result for weddingphotography (MSE: 0.00631)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

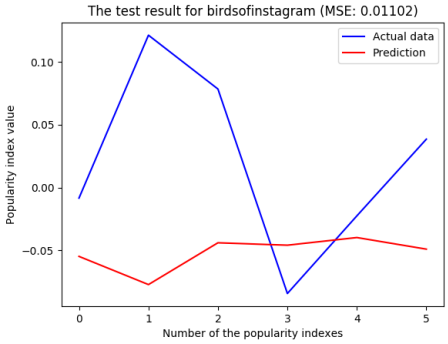

The test result for yorkshire (MSE: 0.03172)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

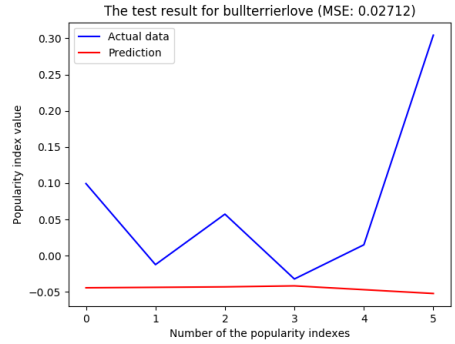
A.2.16 Batch 32, Window 5, Epochs 100

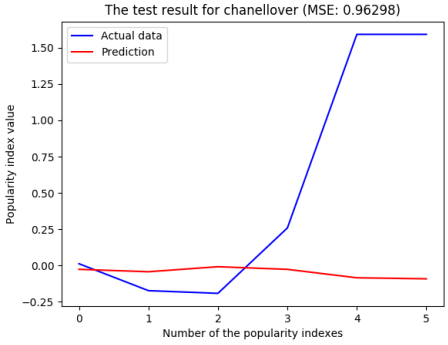


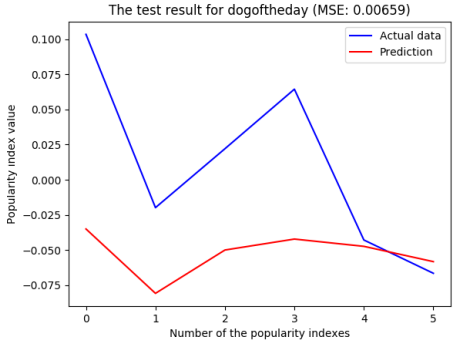


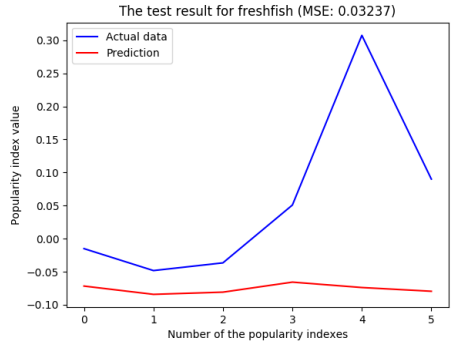


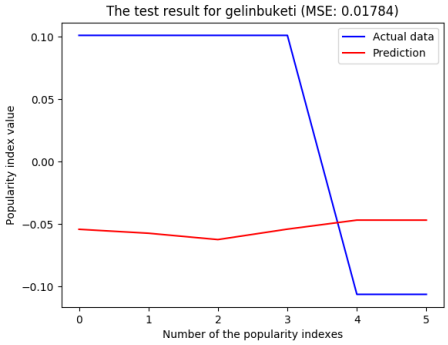


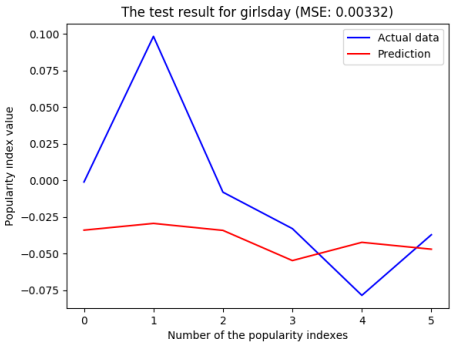


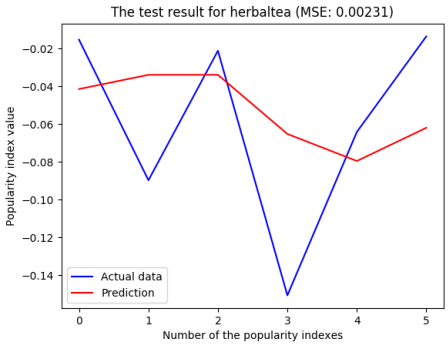


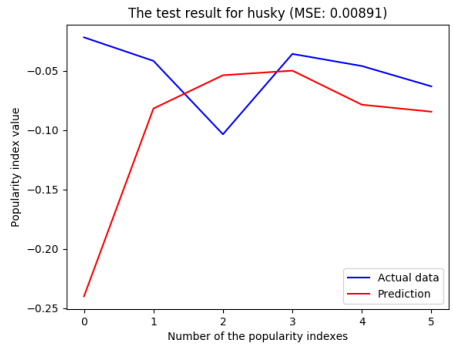


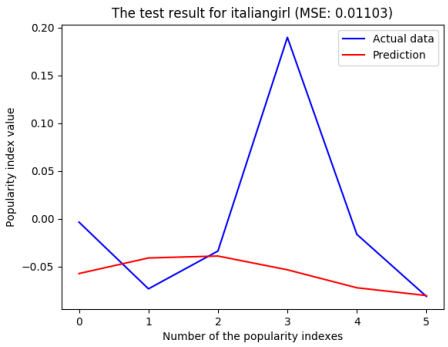

The test result for freshfish (MSE: 0.03237)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

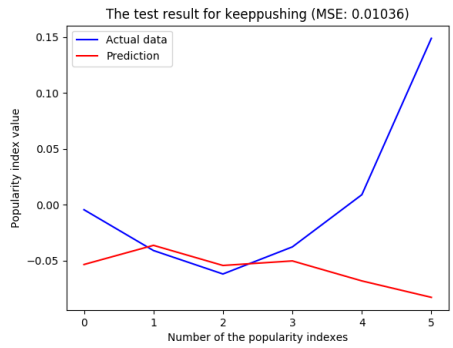

The test result for gelinbuketi (MSE: 0.01784)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

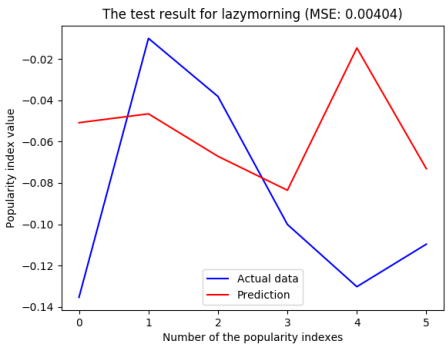

The test result for girlsday (MSE: 0.00332)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

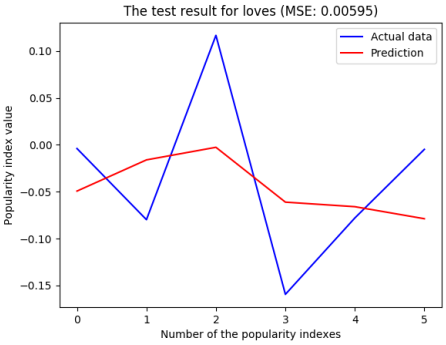

The test result for herbaltea (MSE: 0.00231)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for husky (MSE: 0.00891)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

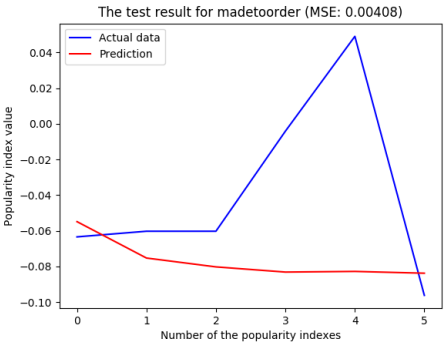

The test result for italiangirl (MSE: 0.01103)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

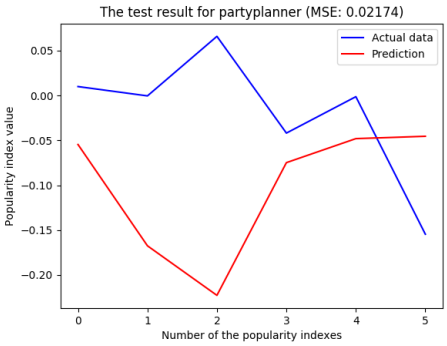

The test result for keeppushing (MSE: 0.01036)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

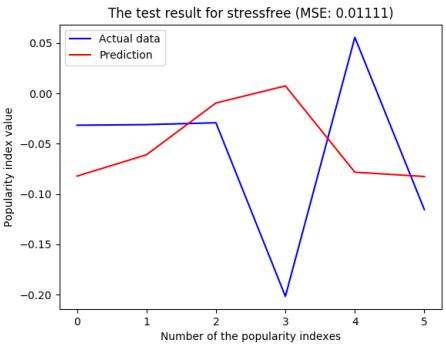

The test result for lazymorning (MSE: 0.00404)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

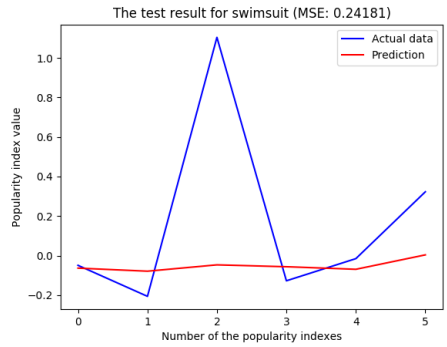

The test result for loves (MSE: 0.00595)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

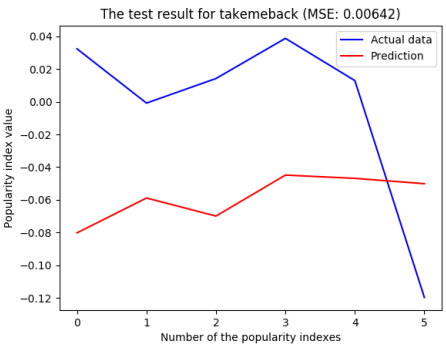

The test result for macncheese (MSE: 0.01450)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

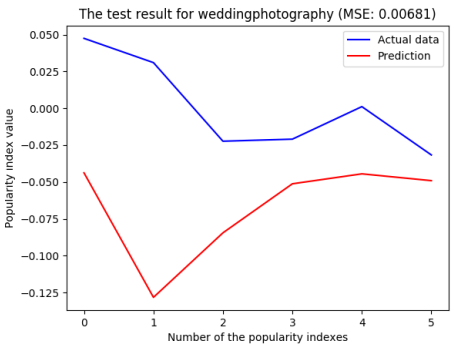

The test result for madetoorder (MSE: 0.00408)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

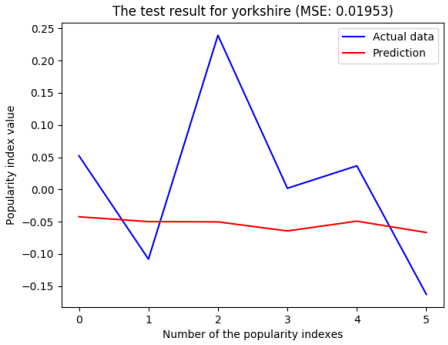

The test result for partyplanner (MSE: 0.02174)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.01111)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

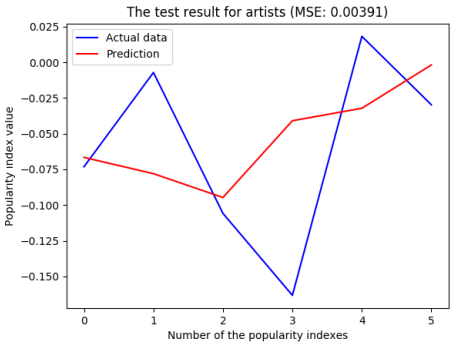

The test result for swimsuit (MSE: 0.24181)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

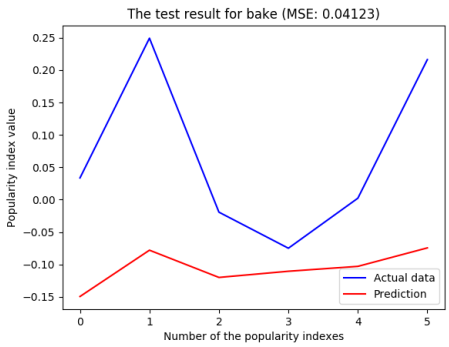

The test result for takemeback (MSE: 0.00642)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

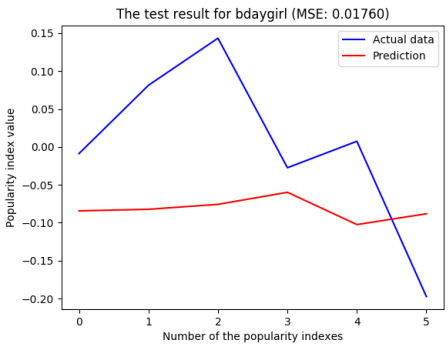

The test result for weddingphotography (MSE: 0.00681)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

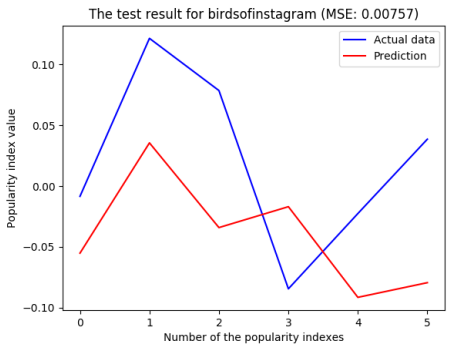


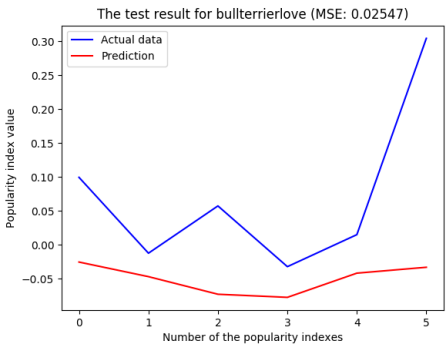
A.2.17 Batch 32, Window 5, Epochs 500

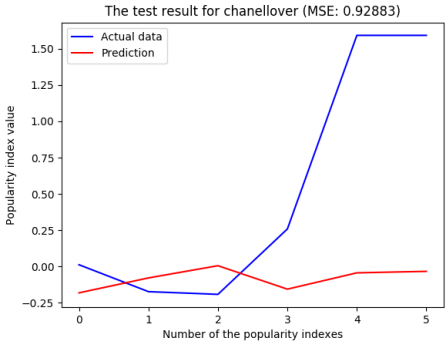


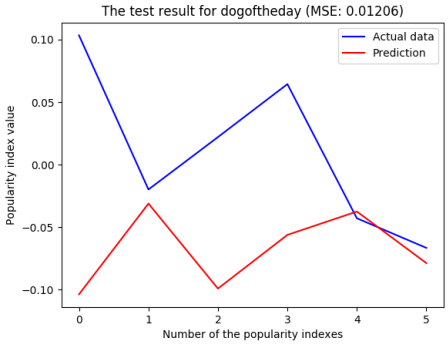


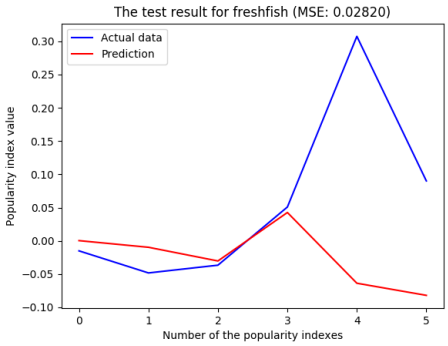


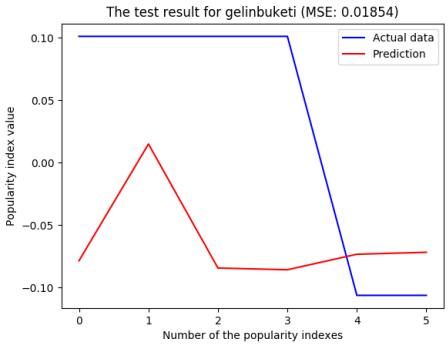


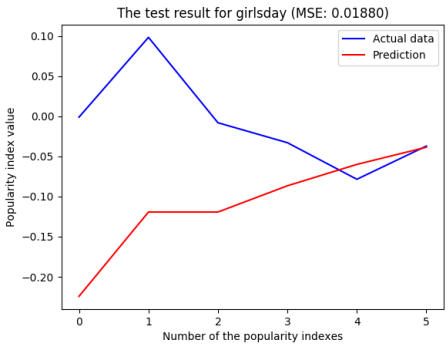


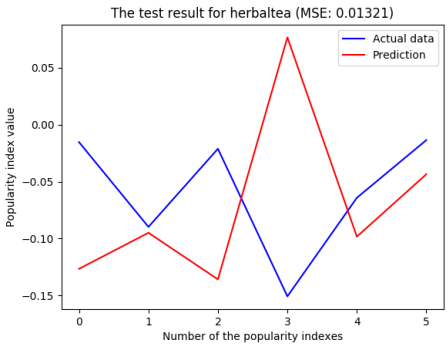

The test result for chanellover (MSE: 0.92883)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

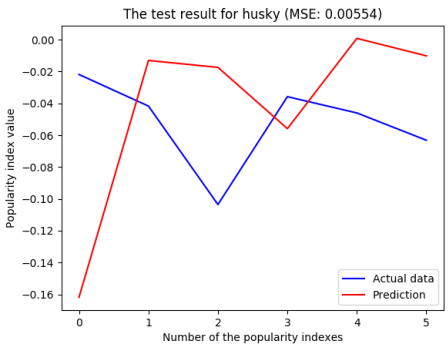

The test result for dogoftheday (MSE: 0.01206)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

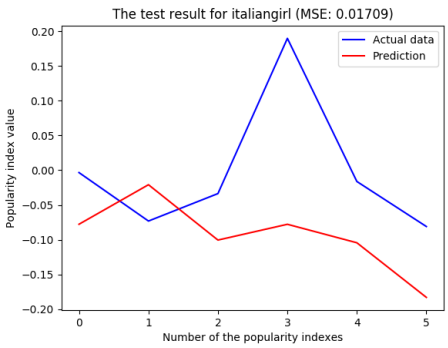

The test result for freshfish (MSE: 0.02820)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

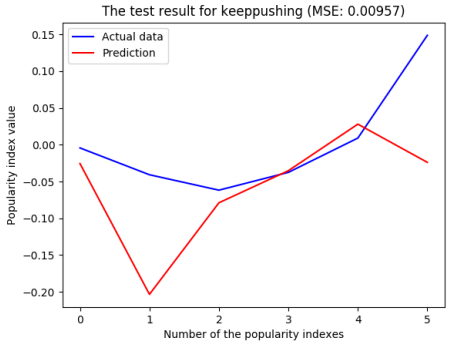

The test result for gelinbuketi (MSE: 0.01854)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

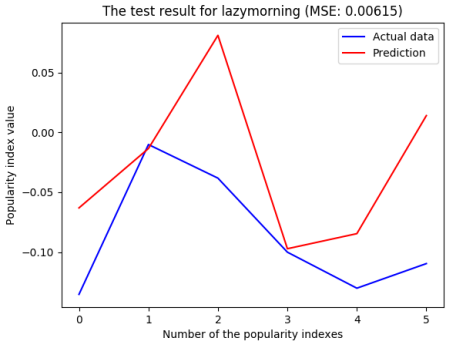

The test result for girlsday (MSE: 0.01880)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

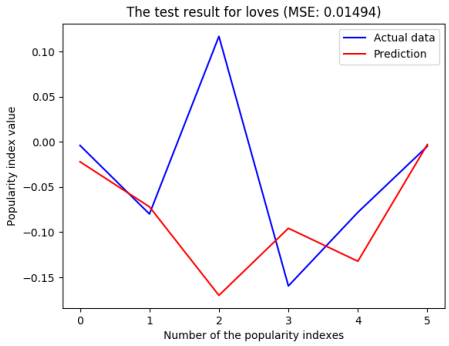

The test result for herbaltea (MSE: 0.01321)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

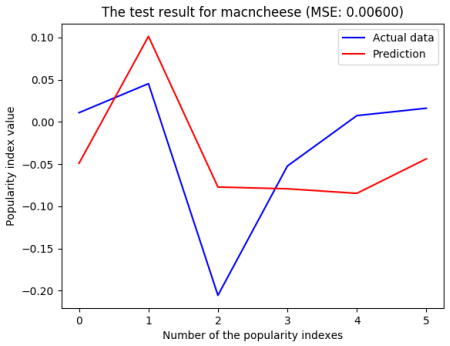

The test result for husky (MSE: 0.00554)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

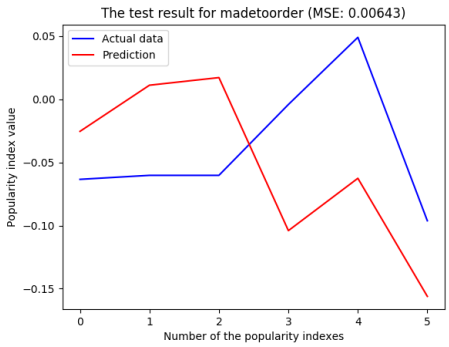

The test result for italiangirl (MSE: 0.01709)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

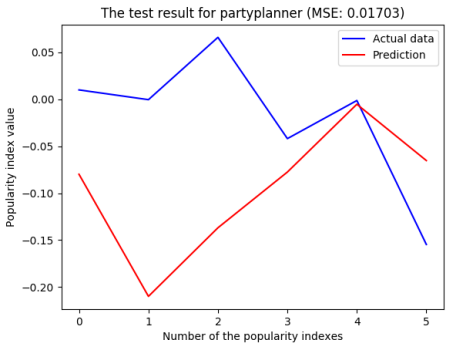

The test result for keeppushing (MSE: 0.00957)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

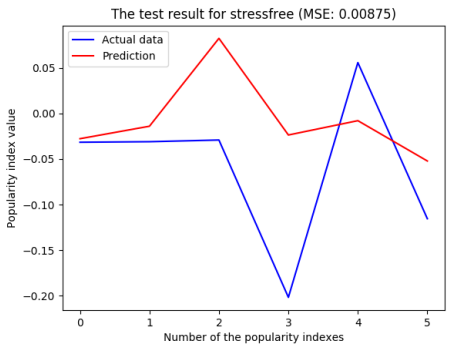

The test result for lazymorning (MSE: 0.00615)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

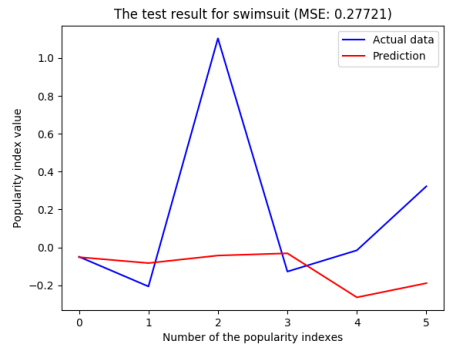

The test result for loves (MSE: 0.01494)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

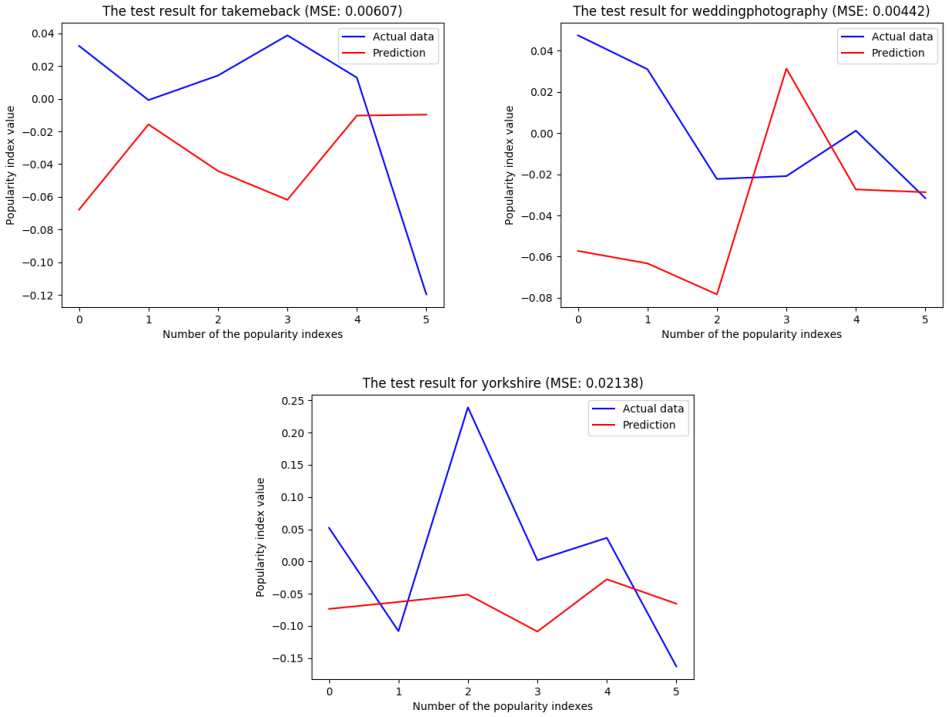

The test result for macncheese (MSE: 0.00600)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for madetoorder (MSE: 0.00643)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

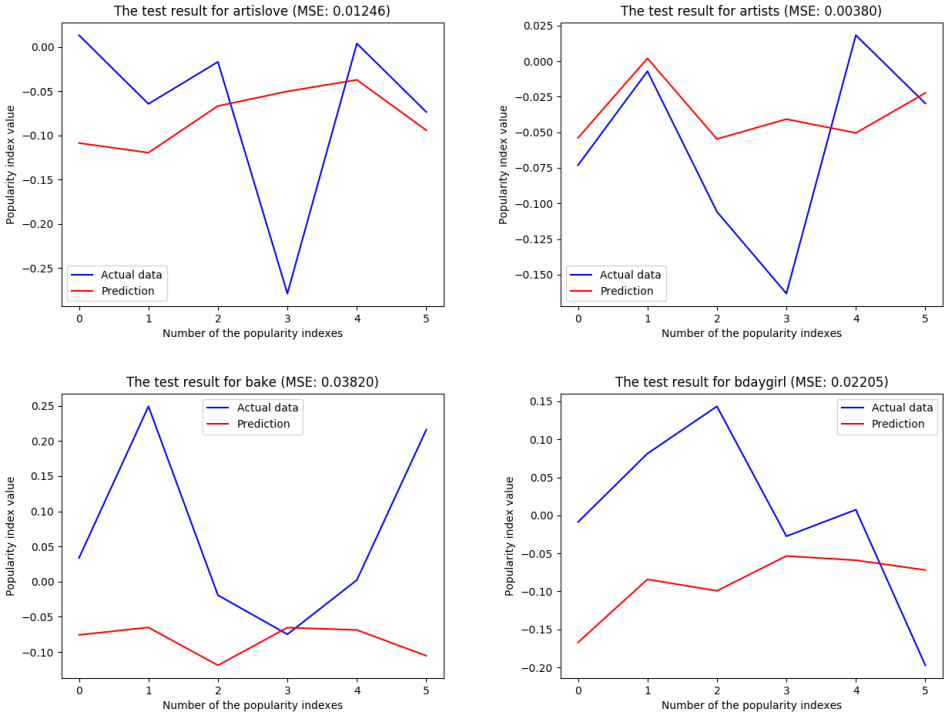

The test result for partyplanner (MSE: 0.01703)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

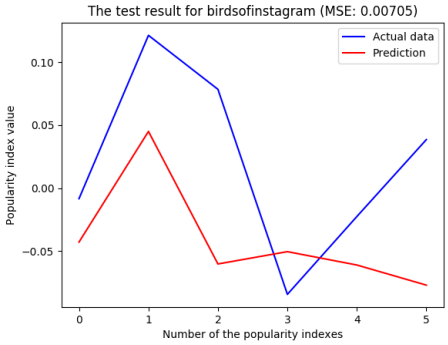

The test result for stressfree (MSE: 0.00875)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

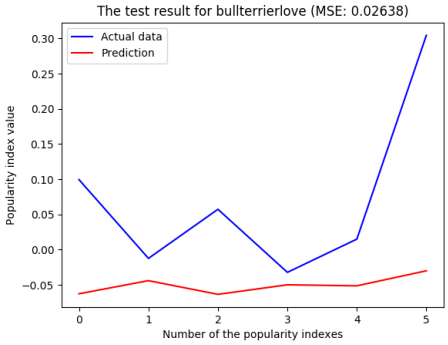

The test result for swimsuit (MSE: 0.27721)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

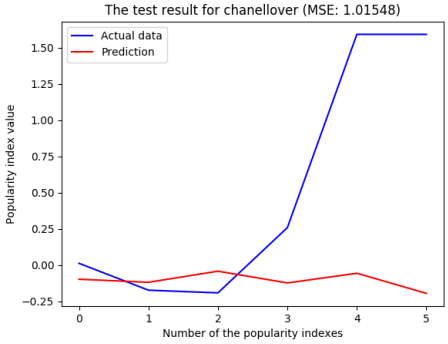


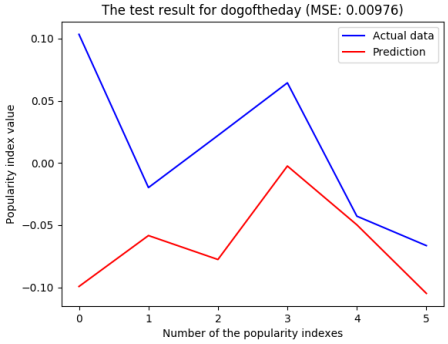
A.2.18 Batch 32, Window 5, Epochs 1000

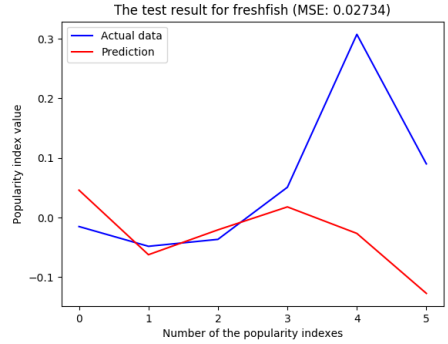


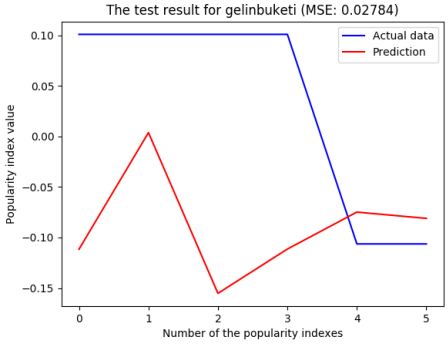

The test result for birdsofinstagram (MSE: 0.00705)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

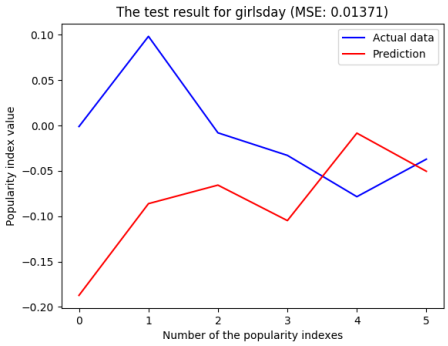

The test result for bullterrierlove (MSE: 0.02638)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

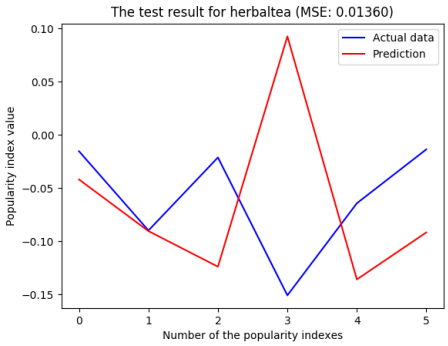

The test result for chanellover (MSE: 1.01548)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

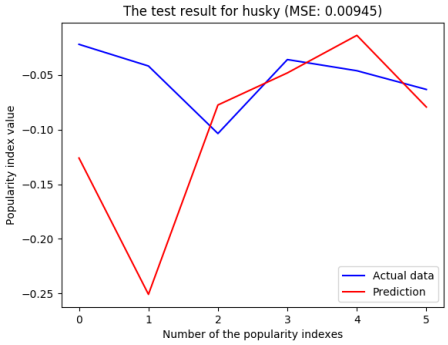

The test result for dogoftheday (MSE: 0.00976)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

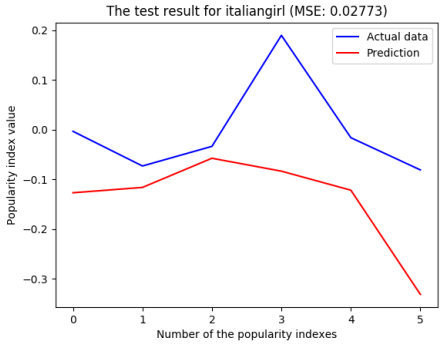

The test result for freshfish (MSE: 0.02734)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

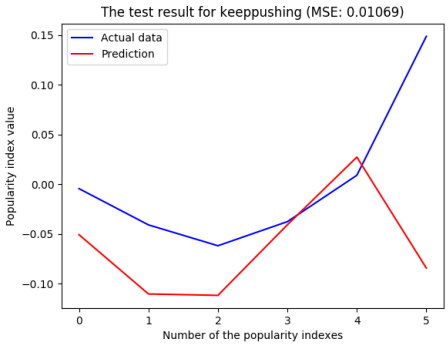

The test result for gelinbuketi (MSE: 0.02784)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

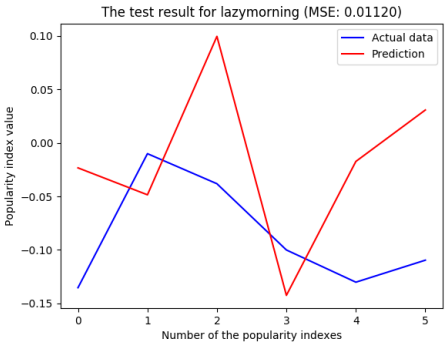

The test result for girlsday (MSE: 0.01371)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for herbaltea (MSE: 0.01360)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

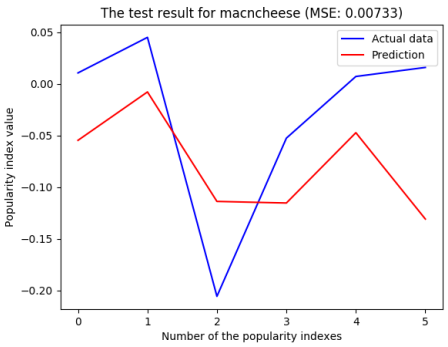

The test result for husky (MSE: 0.00945)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

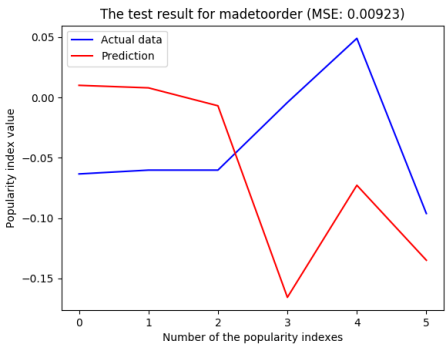

The test result for italiangirl (MSE: 0.02773)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

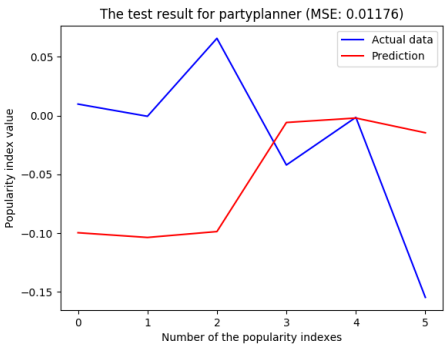

The test result for keeppushing (MSE: 0.01069)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for lazymorning (MSE: 0.01120)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

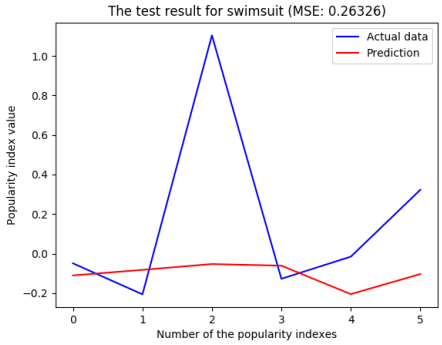

The test result for loves (MSE: 0.00821)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

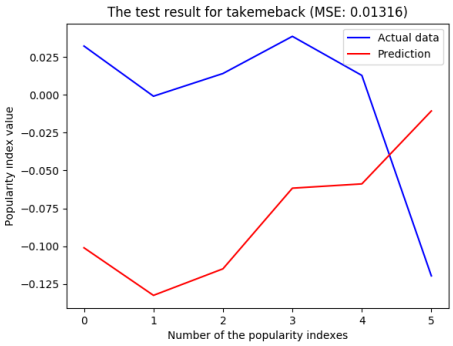

The test result for macncheese (MSE: 0.00733)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

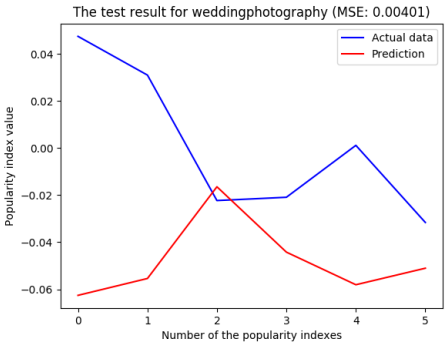

The test result for madetoorder (MSE: 0.00923)
Actual data
Prediction
Popularity index value
Number of the popularity indexes

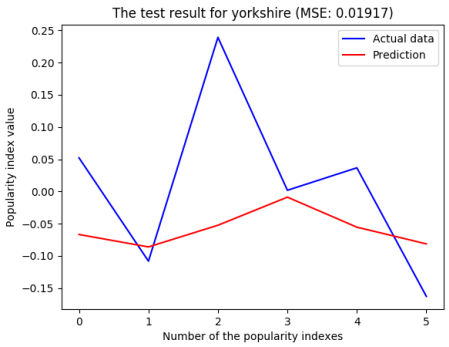

The test result for partyplanner (MSE: 0.01176)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for stressfree (MSE: 0.00722)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for swimsuit (MSE: 0.26326)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for takemeback (MSE: 0.01316)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for weddingphotography (MSE: 0.00401)
Actual data
Prediction
Popularity index value
Number of the popularity indexes


The test result for yorkshire (MSE: 0.01917)
Actual data
Prediction
Popularity index value
Number of the popularity indexes